Fitting signal processing into CNNs with applications to CT denoising

Fitting signal processing into CNNs with applications to CT denoising

Luis Albert Zavala Mondragón

Fitting signal processing into CNNs with applications to CT denoising

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus, prof.dr. S.K. Lenaerts, voor
een commissie aangewezen door het College voor Promoties, in het
openbaar te verdedigen op donderdag 21 december 2023 om 16:00 uur

door

Luis Albert Zavala Mondragón

geboren te Naucalpan de Juárez, México

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ir. M. Mischi
1e promotor:	prof.dr.ir. P.H.N. de With
copromotor:	dr.ir. F. van der Sommen
leden:	prof.dr. I. Išgum (Universiteit van Amsterdam, AMC)
	prof.dr. K.J. Batenburg (Universiteit van Leiden)
	prof.dr.ir. F.M.J. Willems
	prof.dr.ir. C.H. Slump (Universiteit Twente)
adviseur:	dr. K.J. Engel (Philips)

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

To Marjolein, who listens to all my complaints

Fitting signal processing into CNNs with applications to CT denoising

Luis Albert Zavala Mondragón

Cover photo: Brain CT image extracted from the low dose CT image and projection dataset from the Cancer Imaging Archive (Subject no. N079).

Cover design: Karen Zavala Mondragón.

Printed by: ADC Nederland.

ISBN 978-90-386-5918-3
NUR-code 984

Summary

Fitting signal processing into CNNs with applications to CT denoising

Deep neural networks (DNNs) have faced an exponential increase in data analysis applications and modelling performance. Among other factors, the success of these models is driven by the large availability of data and computing power. The performance growth of these models as well as new striking applications have enabled that DNNs and deep learning (DL) go beyond academic and industrial environments and have now a widespread adoption by the general public in specific daily life applications. Besides the mainstream DL applications provided by text and image generation, DNNs have been applied to other fields such surveillance, autonomous driving and (most relevant for this thesis) medical imaging. With the advent of deep learning, conventional signal/image processing algorithms have been often outperformed and replaced by encoding-decoding (ED) CNNs in tasks such as image denoising.

As with all deep learning applications, significant research efforts have been made to enhance the architectures of CNNs. In many cases, these innovations are based on heuristics that offer restricted understanding of the internal operation of these models. In critical applications, such as computed tomography (CT) imaging, it is very important to use well-understood and reliable systems. These requirements have sparked new developments and theoretical insights that explain CNNs from a signal processing perspective, which opens up new exciting opportunities for understanding and improving CNNs.

In order to further advance the understanding in the signal processing behavior of ED CNNs, this thesis builds upon the theoretical insights of the theory of deep convolutional framelets to further strengthen the connections between the conventional signal processing principles and ED CNNs. Furthermore, this thesis applies the developed modelling insights to the noise reduction CNNs for improving the visualization of CT images as an exemplary application.

The start of this thesis research in Chapter 3, discusses and uses signal processing concepts such as framelets, singular value decomposition and the theory of deep convolutional framelets. Based on these elements, Chapter 3 presents an extensive analysis of the encoder-decoder architecture and extends the theory of deep convolutional framelets, by complementing it with concepts such as statistical estimators and Wiener filtering. The explored concepts are used to understand and analyze the signal propagation of CNNs. These analyses results in approaches to modify CNNs such that they become suited for (almost) lossless signal representations, while the non-linear part of the model suppresses the noise. These findings form the basis for modelling contributions in the following chapters.

Chapter 4 builds upon the findings from Chapter 3 to design noise reduction CNNs that are used to enhance low-dose CT images. Specifically, two CNN models, referred to as the *dual-Haar shrinkage networks* 1 and 2 (addressed as *DHSN1* and *DHSN2*, respectively) are presented. These models employ (non-trainable) tight framelets to decompose and reconstruct the signal at multiple scales. To this end, a new framelet transformation based on the Haar basis and on the dual-tree complex wavelet transform is proposed. The developed framelet is referred to as the overcomplete Haar wavelet transform (OHWT) and it has compact and directional filters. Afterwards, this framelet is employed as encoding-decoding structure in two CNN models. Specifically, the OHWT performs a multi-scale decomposition of the input signal. At each of these scales, a Neigh-Shrink-inspired CNN is employed (a strategy to discriminate noise from signal based on the energy of neighbouring pixels) to eliminate the noise contained in the signal. It should be noted that the use of a tight framelet basis circumvents the trainable convolution filters in conventional CNNs, while the Neigh-Shrink-inspired CNN is only able to suppress samples in the detail bands. This clearly contributes to the understanding of the model operation, which contrasts with conventional CNNs such as the filtered back-projection network (FBPConvNet) and the residual encoder-decoder (RED) CNN where the internal operation is unknown.

As a complement to the previous design, the research of Chapter 5 develops a CNN where the encoding and decoding paths are learned for improved flexibility. However, this model distinguishes from other CNNs because it has linear encoding-decoding (ED) paths, while non-linearities are placed on the skip-connection. This means that the proposed design mimics existing wavelet-based denoising algorithms. The simple structure of this system allows to study its linear and non-linear behavior independently, which yields to detailed understanding of its internal operation. Furthermore, the analysis of the linear part of the network proves that the ED path of this model decomposes and reconstructs the signal almost perfectly, while its non-linear part suppresses the noise. Finally, it is worth mentioning that the LWFSN model presented in Chapter 5 employs only a fraction of the training parameters of conventional CNNs (< 1%) and it has a very short inference time (more than 7 times faster than the reference tight-frame U-Net). Despite these simplifications, the LWFSN performs similar to state-of-the-art alternatives, such as the tight frame (TF) U-Net and FBPConvNet for denoising low-dose CT images.

Chapter 6 is more application-oriented and employs the concepts from preceding chapters, by incorporating a framelet-based CNN into a model that decomposes dual-energy (DE) cone-beam (CB) CT into material-specific images. The proposed model is referred to as *regularized conjugate gradient network* (rCGN) and incorporates framelet-based regularization, CNNs and conjugate gradient least squares (CGLS) optimization. It can be observed that the proposed approach is closer to conventional iterative soft thresholding algorithms (ISTAs) than alternative model-based solutions such as FISTA-Net. This means that the proposed rCGN is more interpretable, since ISTA-like algorithms are the result of concrete assumptions about the signal-generation process. In Chapter 6, the rCGN and FISTA-Net are trained with simulated DE CBCT scans and evaluated in both, synthetic and clinically-acquired images. The results in this chapter show that the proposed rCGN uses only 66% of the execution time of FISTA-Net, while achieving comparable peak signal-to-noise ratio and mean structural similarity index when evaluated with synthetic scans. However, the proposed rCGN generalizes better when evaluated with clinically-acquired scans.

In conclusion, this thesis presents an approach to understand CNNs that can be applied to design new CNNs where the internal operation is more interpretable and can be optimized. In addition, the work analyzes the reconstruction characteristics of CNN models, while considering the implications of their linear and non-linear parts. Consequently, it is shown that such analyses reveal internal conditions that the models should comply with in order to preserve the signal. Moreover, this dissertation is among the first in exploring the signal reconstruction behavior of trained ED CNN models and confirm it by a theoretical analysis.

Samenvatting

Fitting signal processing into CNNs with applications to CT denoising

De toepassingen en prestaties van diepe neurale netwerken (DNN's) zijn exponentieel toegenomen voor onder andere het analyseren van gegevens en het modelleren van beelddata. Deze modellen hebben hun succes onder andere te danken aan de grote beschikbaarheid van beelddata en toegenomen rekenkracht. De prestatiegroei van deze modellen alsmede de nieuwe interessante toepassingen, hebben ervoor gezorgd dat DNN's en diep leren (*deep learning*) (DL) ook buiten de academische en industriële omgevingen optreden en nu op brede schaal door het grote publiek worden gebruikt voor toepassingen in het dagelijks leven. Naast de hoofdtoepassingen voor DL, zoals het genereren van tekst en afbeeldingen, zijn DNN's ook toegepast op andere gebieden, zoals surveillance, autonoom rijden en (het meest relevant voor dit proefschrift) medische beeldvorming. Met de komst van DL worden de conventionele algoritmes voor signaal- en beeldverwerking vaak overtroffen en vervangen door encoding-decoding (ED) Convolutionele NN's (CNN's) bij bekende taken zoals ruisonderdrukking in beelden.

Zoals bij alle DL-toepassingen is er veelvuldig onderzoek gedaan om de architecturen van CNN's te verbeteren. In veel gevallen zijn deze innovaties gebaseerd op heuristische aanpakken die slechts een beperkt begrip bieden van de interne werking van deze modellen. In kritieke toepassingen, zoals computertomografie (CT), is het erg belangrijk om betrouwbare systemen te gebruiken, waarvan de werking goed begrepen wordt. Dit inzicht heeft geleid tot nieuwe ontwikkelingen en theoretische inzichten die CNN's verklaren vanuit een signaalverwerkingsperspectief, wat zorgt voor nieuwe interessante mogelijkheden om CNN's te begrijpen en te verbeteren.

Om meer inzicht te krijgen in het signaalverwerkingsgedrag van ED CNN's, bouwt dit proefschrift voort op de theoretische inzichten op het gebied van *deep convolutional framelets*, een raamwerk om inverse problemen bij (perfecte) beeldreconstructie beter op te lossen. Deze aanpak blijkt in staat om de verbanden tussen de conventionele signaalverwerkingsprincipes en ED CNN's verder te versterken. Verder worden in dit proefschrift, als voorbeeld van een mogelijke toepassing, de ontwikkelde modelleringsinzichten toegepast op de ruisonderdrukkende CNN's voor het verbeteren van de visualisatie van CT-beelden.

In hoofdstuk 3 worden concepten van signaalverwerking zoals *framelets*, decompositie van singuliere waarden en de theorie van *deep convolutional framelets* uitgelegd. Op basis van deze elementen presenteert hoofdstuk 3 een uitgebreide analyse van de ED-architectuur en breidt de theorie uit van *deep convolutional framelets* door deze aan te vullen met concepten als statistische schatters en Wiener-filtering. De onderzochte concepten worden gebruikt om de signaalpropagatie in CNN's te begrijpen en te analyseren. Deze analyses resulteren in benaderingen om CNNs zodanig aan te

passen dat ze geschikt worden voor bijna verliesvrije signaalrepresentaties, terwijl het niet-lineaire deel van het model de ruis onderdrukt. Deze bevindingen vormen de basis voor diverse modelleringsbijdragen in de volgende hoofdstukken.

Hoofdstuk 4 bouwt voort op de bevindingen van hoofdstuk 3 om ruisonderdrukkende CNN's te ontwerpen die worden gebruikt om CT-beelden te verbeteren, die met lage stralingsdosis opgenomen zijn. Specifiek worden twee CNN-modellen gepresenteerd, die de *Dual-Haar Shrinkage Networks* 1 en 2 worden genoemd (respectievelijk DHSN1 en DHSN2). Deze modellen maken gebruik van (niet-trainbare) compacte *framelets* om het signaal op verschillende schaalgroottes te ontleden en te reconstrueren. Hiertoe wordt een nieuwe *framelet*-transformatie voorgesteld die gebaseerd is op de Haar-basis en op de *dual-tree* complexe *wavelet*-transformatie. De ontwikkelde *framelet* wordt de Overcomplete Haar Wavelet Transformatie (OHWT) genoemd en heeft compacte en directionele filters. Daarna wordt deze *framelet* gebruikt als ED-structuur in twee CNN-modellen. De OHWT voert een decompositie uit op meerdere schaalgroottes van het ingangssignaal. Op elk van deze schaalfactoren wordt een op *Neigh-Shrink*-geïnspireerde CNN gebruikt (een strategie om ruis van het signaal te onderscheiden op basis van de energie van naburige pixels) om de ruis in het signaal te elimineren. Opgemerkt moet worden dat het gebruik van een compacte *framelet*basis de trainbare convolutiefilters in conventionele CNN's omzeilt, -terwijl de op *Neigh-Shrink*-geïnspireerde CNN alleen signalen in de detailbanden kan onderdrukken. Dit aspect draagt duidelijk bij aan het begrijpen van de werking van het model, in tegenstelling tot conventionele CNN's, zoals het *filtered back-projection network* (FBPConvNet) en de *residual encoder-decoder (RED)* CNN, waarvan de interne werking onbekend is.

Als aanvulling op het vorige ontwerp, ontwikkelt het onderzoek van hoofdstuk 5 een CNN waarbij de coderings- en decoderingspaden worden aangeleerd voor verbeterde flexibiliteit. Dit model onderscheidt zich echter van andere CNN's doordat het lineaire coderings- en decoderingspaden (ED) heeft, terwijl niet-lineariteiten worden geplaatst op de *skip*-connecties in het netwerk. Dit betekent dat het voorgestelde ontwerp reeds bestaande, op *wavelet*-gebaseerde ruisonderdrukkingsalgoritmen nabootst. De eenvoudige structuur van dit systeem maakt het mogelijk om het lineaire en niet-lineaire gedrag onafhankelijk van elkaar te bestuderen, wat leidt tot een gedetailleerd begrip van de interne werking. Bovendien bewijst de analyse van het lineaire deel van het netwerk dat het ED-pad van dit model het signaal bijna perfect ontleedt en reconstrueert, terwijl het niet-lineaire deel de ruis onderdrukt. Tot slot is het zinvol om te vermelden dat het LWFSN-model uit hoofdstuk 5 slechts een fractie van de trainingsparameters van conventionele CNN's gebruikt (< 1%) en dat het een zeer korte executietijd heeft (meer dan 7 keer sneller dan de referentie-*tight-frame* U-Net). Ondanks deze vereenvoudigingen presteert het LWFSN vergelijkbaar met alternatieven uit de literatuur, zoals het *tight-frame* (TF) U-Net en FBPConvNet voor de ruisonderdrukking van CT-beelden met een lage stralingsdosis.

Hoofdstuk 6 is meer toepassingsgericht en maakt gebruik van de concepten uit voorgaande hoofdstukken door een *framelet*-gebaseerde CNN op te nemen in een model dat *dual-energy (DE) cone-beam (CB)* CT ontleedt in materiaal-specifieke beelden. Het voorgestelde model wordt aangeduid als *geregulariseerd geconjugeerd gradiënt-*

netwerk (rCGN) en bevat op *framelet*-gebaseerde regularisatie, CNN's en geconjugeerde-gradiënt met kleinste-kwadraten (CGLS) optimalisatie. Er kan worden opgemerkt dat de voorgestelde aanpak dichter bij conventionele iteratieve *soft-thresholding* algoritmen (ISTA's) ligt dan alternatieve modelgebaseerde oplossingen zoals FISTA-Net. Dit betekent dat de voorgestelde rCGN beter interpreteerbaar is, aangezien ISTA-achtige algoritmen het resultaat zijn van concrete aannames over het signaalgeneratieproces. In hoofdstuk 6 worden de rCGN en FISTA-Net getraind met gesimuleerde duale-energie CBCT-scans en geëvalueerd met zowel synthetische als klinisch verkregen beelden. De resultaten in dit hoofdstuk laten zien dat het voorgestelde rCGN slechts 66% van de executietijd van FISTA-Net gebruikt, terwijl het een vergelijkbare pieksignaal-ruisverhouding en gemiddelde structurele overeenstemmingsindex (SSIM) bereikt voor synthetische scans. De voorgestelde rCGN generaliseert echter beter wanneer deze wordt geëvalueerd met klinisch verkregen beeldopnamen.

Concluderend presenteert deze dissertatie een invalshoek om CNN's beter te begrijpen, die kan worden toegepast om nieuwe CNN's te ontwerpen waarbij de interne werking veel beter interpreteerbaar is en kan worden geoptimaliseerd. Bovendien analyseert het werk de reconstructiekenmerken van CNN-modellen, waarbij de implicaties van hun lineaire en niet-lineaire delen afzonderlijk in overweging worden genomen. Er wordt aangetoond dat dergelijke analyses interne voorwaarden laten zien waaraan de modellen moeten voldoen om de signaalkwaliteit te behouden. Bovendien is het werk uit dir proefschrift een van de eerste onderzoeken waarin het signaalreconstructiegedrag van getrainde ED-CNN-modellen wordt onderzocht en bevestigd door een theoretische analyse.

Contents

CHAPTER 1

Introduction

1.1 Current context of deep learning

Fueled by the development of deep learning (DL), deep neural networks (DNNs) have faced an exponential increase in applications and performance. Among other factors, the success of these models is driven by the large availability of data and computing power. The performance growth of these models as well as new striking applications have enabled that DNNs and DL go beyond academic and industrial environments and have now a widespread adoption by the general public. For example, large language models (LLMs) such as *ChatGPT* [1], *Dall-E2* [2], *Imagen* [3] or the *segment-anything model* (SAM) [4] are all popular models that are employed by millions of users worldwide.

Besides the mainstream DL applications provided by text and image generation with LLMs, DNNs have been applied to other fields such surveillance, autonomous driving and (most relevant for this thesis) medical imaging. It should be noted that for all these applications, many lower-level vision tasks are used. Examples of these tasks are image recognition/classification [5], object detection [6], [7], image segmentation [8], image compression and image denoising [9]. Prior to DL, many computer vision and image processing tasks were performed with conventional machine learning, e.g. Support Vector Machines [10], Random Forests [11], Haar cascades [12], as well as image/signal processing-based solutions, such as total-variation minimization [13], wavelet-based algorithms [14], sparse coding [15], etc. It can be observed that the previously mentioned techniques perform assumptions/modeling about the signal nature.

As an insightful example of the above mentioned modeling. The famous Viola-Jones [12] algorithm for face and object detection integrates conventional signal processing and machine learning processes for the task of face/object detection. That is, the Viola-Jones algorithm uses a set of Haar-like filters which are correlated with face attributes. These filters, are efficiently convolved with the image pixels with the use of integral images. Finally, the resulting feature maps are supplied as input to a machine learning classifier [16] to generate object/face candidates. Finally, In order to remove spurious detections, the so-called *attention cascade*, is employed.

With the advent of deep learning, the use of conventional signal/image processing and machine learning algorithms have been outperformed and replaced by DNNs.

Consequently, the explicit signal modeling of well-understood algorithms has been exchanged by general-purpose DNNs that are re-trained for specific tasks. For example, the multiple stages of the Viola-Jones algorithm (feature extraction, classification/detection, elimination of spurious detections) can be replaced by a single DNN model, such as the popular object detection model YOLO [7], which performs all the aforementioned tasks of Viola-Jones algorithm in a single end-to-end solution.

Despite of the high-performance and simplicity of training DNNs, their training and deployment have disadvantages that should be considered. For example, DNNs have high model complexity [17], they can be easily misled by adversarial attacks [18], they are prone to overfitting and their internal operation is often unknown [19]. In addition, it has been found that producing marginal gains in performance comes at the cost of an exponential increase in computational complexity [20]. Finally, it should be noted DNNs require specialized, expensive and power-hungry hardware for training purposes. This presents a challenge for low-power systems which are embedded in daily-life appliances [21], as well as computation-constrained devices. This discussion even raises environmental concerns due to the required power consumption which incurs a high-carbon footprint, which is directly linked to train and deploy these models. For example, the training of the GPT-3 model (one of the core elements of the well-known ChatGPT) consumes more than 1000 MWh of electricity. Producing this amount of electricity releases approximately the same amount of CO_2 as 500 round flights from New York to Paris [22], [23]. In addition, it should be considered that the ever-growing computational demands of DNNs require the continuous renovation of computer hardware, which translates into more electronic waste and more demand for raw materials.

As closing argument for this section, it should be noted that further challenges can be associated to DNNs employed in critical applications such as medical imaging. For example, in this thesis DNNs are employed to improve the quality of computed tomography (CT) images. For this application, the DNN should preserve the information contained in the image. Furthermore, for such medical applications, additional properties such as robustness or explainability may be desirable. It should be noted that for these aspects, the black-box nature of common DNNs presents a challenge, since it is difficult to provide any of the described properties with current DNN models. The following section addresses in more detail the formation and meaning of CT scans and the challenges of DNNs within this context.

1.2 Computed tomography imaging

X-ray-based computed tomography (CT) is an imaging modality, where the generated images display the X-ray attenuation coefficients of the scanned tissues. The volumetric data retrieved by a CT scanner is often grouped into *slices*. The attenuation values occurring in CT images are related to the material composition of the scanned body part. This means that for brain images, gray/white matter, blood, bone other and other tissues have all specific intensities. For example, intracranial hemorrhages display increased brightness because the liquid blood is more attenuating for X-rays

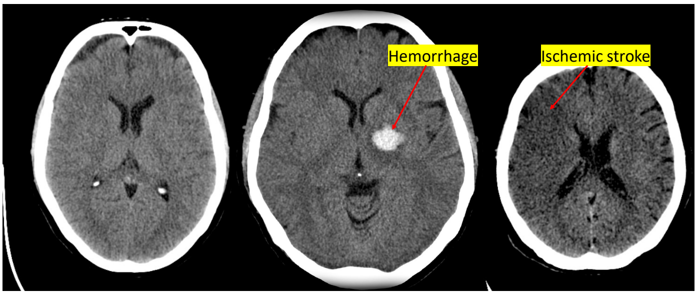


Figure 1.1. Axial slices of head CT scans of a patient without (left) and with brain lesions (center, right). Note that the center image has an hemorrhage, which is displayed as a bright blob on the image, wile the right image showcases a stroke, that is displayed as a decreased the attenuation in the brain parenchima. Cases courtesy of Dr. David Cuete and of Assoc. Prof. Frank Gaillard, Radiopaedia.org, rIDs: 23768, 2764, 35427, respectively.

than gray or white matter (see Fig. 1.1). On the contrary, acute brain infarcts decrease the X-ray attenuation in the brain parenchima and reduce the differentiation between gray and white matter. It should be noted that in CT imaging, the contrast between the diverse tissues in the brain parenchima is small (typically represented by changes of a few Hounsfield units). Consequently, images aiming for diagnostic purposes require low noise levels, since the noise presence can easily hide lesions within the image, as shown in Fig. 1.2.

A CT acquisition is achieved by *back-projecting* a number of 2D X-ray projections onto a 3D volume. At a high level of discussion, an X-ray projection is acquired as follows.

First, an X-ray source generates photons that propagate through the medium (e.g. air, patient and other objects). During the travel of the photons towards the detector, the X-ray photons may collide with atoms of diverse materials (e.g. bone, water, brain), which can result in a change of trajectory and a loss of energy (Compton effect), or in the absorption of the photon by the atom resulting in the release of electrons by the atom (photoelectric effect). The photoelectric effect is more likely to happen for dense materials. Consequently, the detector will receive more energy in the paths where low-density materials are present (e.g. air, soft tissue). Finally, the energy of the photons that reach the detector during the projection is integrated and the final readout is retrieved.

Second, it should be noticed that the interactions between the photons and the material in the medium are statistical interactions, which means that the photon flux is subject to random variations of the measured intensity (noise). Furthermore, the electronic circuitry of the system also introduces noise in the acquired projections. In X-ray medical imaging, the noise can modeled as a Poisson-distributed random variable, where the variance is the expected number of photons.

Third, after the acquisition of multiple X-ray projections a reconstruction algorithm maps the projections into a volumetric CT image. It is important to consider that the noise contained in the X-ray projections will also affect the reconstructed images, which will present signal-dependent noise, that is also non-stationary and has spatial

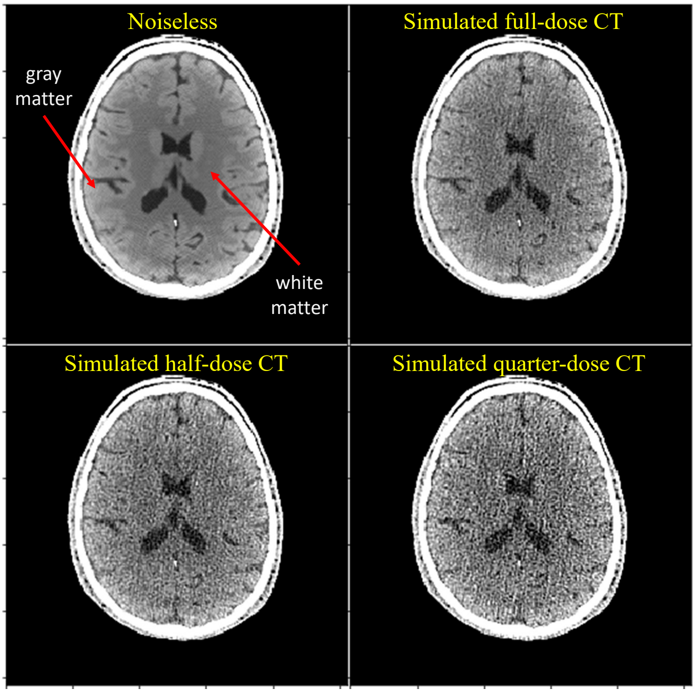


Figure 1.2. Simulated noiseless full-, half- and quarter-dose CT slices. Note that as the noise increases, the radiation dose decreases and it is no longer possible to differentiate the gray and white matter.

correlations. These properties result from the filtering stages that are applied in the reconstruction algorithms.

Because of the inherent noise properties of the detected signal, the handling and treatment of noisy signals requires careful attention. Specifically, higher-quality images with low noise levels are obtained with high-radiation doses. However, it is desirable to reduce the exposure to X-rays because there are direct health risks for the patient and medical personnel [24]. This directly explains why the radiation doses applied in X-ray systems have continuously decreased over the past decades [25], [26]. Other factors that may lead to images with higher noise content are the generation of high-resolution scans [27] and the use of spectral detectors [28]. These factors highlight the relevance of using noise-removal algorithms, while preserving the underlying anatomical structures. Common approaches for noise reduction in CT are total-variation minimization [29], framelet-based methods [30], sparse coding [31], low-rank approximation [32], [33], collaborative filtering [34], [35] and more recently, data-driven methods such deep neural networks [36]–[38].

From the previous discussion, it can be deduced that if noise reduction in CT is achieved with DNNs, the DNN models should not alter any signs of pathology, morphological information, small structures and/or the intensity levels of the picture, since this could alter the medical diagnosis, quantitative measurements and/or subsequent processing. In addition, given the potential patient risks, it is desirable that the models provide some degree of interpretability to clinicians. This last point is critical, because it allows the clinicians to evaluate the capabilities, limitations and possible risks associated to such models. Among the DNN architectures employed in medical imaging, a common solution is the so-called encoding-decoding (ED) convolutional neural network (CNN) architecture. This model is suitable for pixel-level tasks such as image denoising, since it has as input and output images with the same resolution. Such an ED CNN model is described in more detail in the following section.

1.3 Encoding-decoding (ED) CNNs

The so-called encoding-decoding CNNs are models which consist of the following elements. First, the *encoder* that maps the input to a multi-channel signal representation. Second, the *decoder* that maps the encoded signal back to the original domain using a multi-layer approach, which results in an image of the same dimensionality as the original input. The third and final element of ED CNNs are the sparsifying non-linearities that are embedded within the encoder and decoder sections and suppress parts of the signal. It should be noted that this last operation is one of the main contributors to the noise reduction behavior of the model.

The potential of encoding-decoding CNNs as an approach for obtaining more interpretable DL-based noise reduction can be related to the fact that the mathematical formulation of ED CNNs has strong similarity to well-established signal processing techniques for noise reduction. For this thesis, it is particularly important to develop the connection to shrinkage-based algorithms in the wavelet domain [39], low-rank [40] approximation and sparse coding [41]. It should be noted that the mathematical formulations of the three aforementioned techniques are a consequence of explicit assumptions about the data and noise model. The choice for this three techniques us further motivated in the following section and will be come gradually clear.

A simple visual example of the similarity between the wavelet transformation and the encoding-decoding structure of a CNN is shown in Fig. 1.3, where it can be directly observed that the implementation of a multi-level discrete wavelet transform with a filter bank [42] is very similar to common ED CNNs, such as the U-Net [8]. This similarity is described by the property that both approaches develop a tree-like structure, which is produced by convolving the incoming data with a convolution kernel. This step can be repeated in order to process the information at multiple resolutions and with more detailed decompositions. One important difference between the two structures is the fact that the filters of wavelet transform are defined in advance so that the information structures for filtering are known, whereas in the U-Net the filtering structures are learned from the data to perform a specific task. It can be considered that we are not the first in reporting such similarity, which has already been described in the theoretical work by Ye *et al.* [40], [43] that is addressed in more detail in subsequent text.

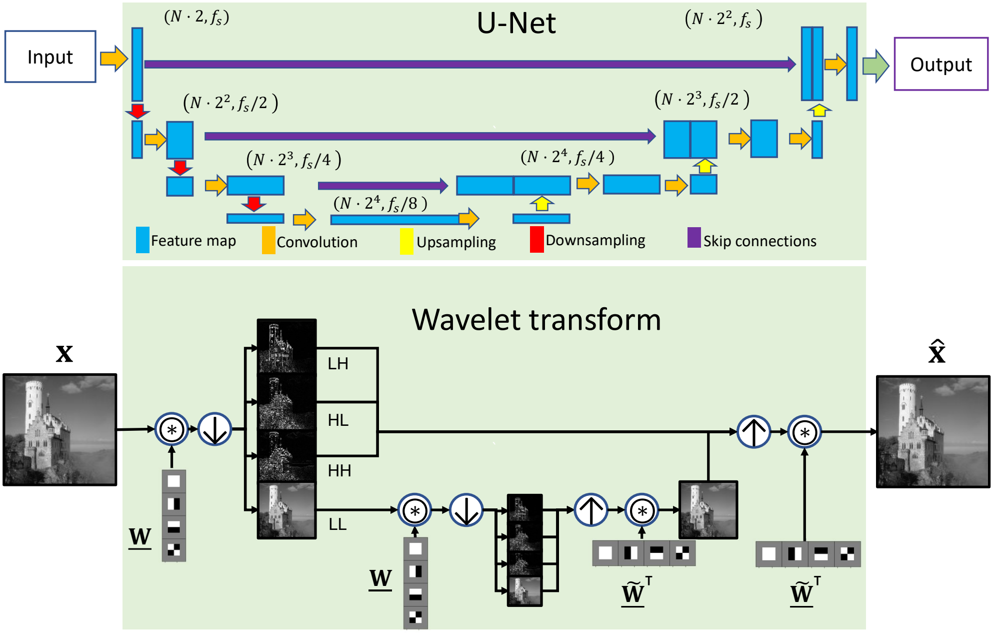


Figure 1.3. Comparison between the architecture of the U-Net and the discrete wavelet transform. In the U-Net diagram, N is the number of feature maps, while f_s is the spatial sampling frequency/feature map resolution. Furthermore, in the diagram corresponding to the wavelet transform, $\underline{\mathbf{W}}$ and $\underline{\tilde{\mathbf{W}}}$ are the basis functions for the forward and inverse transform, respectively. In the figure the up/down arrows correspond to up/down sampling operations, while operator $\circledast$ is a tensor convolution.

Signal processing-based noise reduction. Prior to addressing some of the commonalities between signal processing and ED CNNs, two common approaches for noise reduction which are linked to the theoretical explanations of the operation of noise reduction ED CNNs are addressed. The first approach is wavelet shrinkage-based noise reduction [14], which employs the noise model and/or signal model, as well as intra/inter-scale relationships in the wavelet-transformed domain for eliminating noise [44]. The second signal processing-based approach addressed here is sparse coding [15], employing data to generate a prior distribution of patches (referred to as *atoms*), which are grouped into a *dictionary*. In order to produce an estimate, the noisy signal is approximated by a sparse linear combination of the elements in the dictionary, where the weights that are applied for the linear combination of atoms are referred to as *code* in the sparse coding view, whereas in the the neural network domain they are called hyper-parameters.

Differences between signal processing approaches and encoding-decoding CNNs. In contrast with the previously addressed signal processing denoising algorithms, encoding-decoding CNNs are often presented as a solution without making explicit assumptions on the signal and noise. For example, in supervised algorithms, an encoding-decoding

CNN *learns* the optimal parameters to filter the signal from a set of paired examples of noise/artifact-free images and images contaminated with noise/artifacts [9], [36], [37]. This data-driven approach highly simplifies the solution of the noise-reduction problems, since this circumvents the use of explicit modeling of the signal and noise. Still, it should be noted that the added flexibility comes at the expense of lacking the understanding of the internal operation, which is typically discussed in model-centered designs [21].

The vision. It can be observed that the commonalities between ED CNNs and sparse transforms such as the discrete wavelet transform present an opportunity to introduce explainability in ED CNNs. For example, the noise is often considered to be a random signal with high-frequency components, while most of the content in natural images is often containing mostly low-frequency energy. Consequently, it could be possible to leverage the space-frequency partitioning properties of discrete wavelet transformations to provide a suitable representation, such that the high-frequency information is separated from the low-frequency part and to apply a CNN that only eliminates noise in the high-frequency segment of the signal. It can be observed that, by constraining the operation of the CNN to a specific task, the model becomes by definition less of a "black box". Furthermore, by constraining the CNN to operate in the high-frequency bands, it is avoided that critical components in the low-frequency band are processed. Alternatively, it is possible to explore how a CNN, where the encoding-decoding path is learned, could be designed so that it behaves more akin to conventional signal processing algorithms. Hence, by mimicking a known algorithm, the model becomes more explainable as well.

The following section addresses in more detail the approaches and challenges faced to achieve interpretable CNNs in the field of medical imaging. Furthermore, Section 1.4 also addresses the existing theoretical frameworks which explain the structure of ED CNNs from a signal processing perspective.

1.4 Challenges of CNNs for noise reduction in CT

Previous sections have addressed the relevance of achieving interpretable operations of CNNs for critical applications such as medical imaging, as well as our vision of increasing the interpretability of CNNs by integrating well-understood signal processing and machine learning concepts. This section further details this discussion and breaks it up in several view points for detailing advances in existing literature.

Hybrid deep learning for increased interpretability. A basic point to start with is the combination of machine learning/deep learning and signal processing concepts as a hybrid engine for designing explainable CNNs. For example, well-established models such as decision trees have been explored as a tool to understand general CNNs [45], [46]. Another important example of hybrid models is the recent work by Khozeimeh *et al.* [47], who proposed a hybrid model which contains convolutional features as feature extractors and a decision tree as classifier. In the realm of image restoration, the integration of signal processing concepts have also resulted in new models that improve the performance of existing CNNs. For example, the work by Han *et al.* [40] demonstrates the limitations of the conventional pooling structure used in the U-Net [8] and instead improves upon this shortcoming by using the wavelet transform to avoid the loss of high-frequency information. On the other hand, the models of Mentl *et al.* [48] and Fan *et a.* [39] introduce shrinkage-based networks in CNNs that are used for denoising of CT scans. It can be noted that despite of the already impressive achievements of these models, they also have some limitations. For example, none of these methods explore the properties of the convolutional layers of the design, while the placement of the non-linearities in the model is regularly not justified based on any signal properties and/or assumptions.

Theoretical connections between encoding-decoding CNNs and signal processing. In the specific context of signal processing tasks, in recent years, there have been significant efforts to improve the understanding of the internal operation of CNNs, by comparing the internal operation of ED CNNs with signal processing concepts. For example, the theory of deep convolutional framelets (TDCF) [40] explores the behavior of ReLU CNNs and establishes that the linear structure of CNNs is akin to a data-driven framelet decomposition. In addition, the TDCF also states that the denoising behavior is a similar to low-rank approximation, which is achieved CNNs by suppressing feature maps/signal bases with the ReLUs. An alternative approach is the sparse coding interpretation [41], which describes the operation of CNNs from a convolutional sparse-coding approach, where the output of a given neighborhood is a sparse combination of learned basis functions. Finally, the spline modeling of CNNs, mentions that the output of ReLU-based CNNs can be considered as first-order spine approximations of the noisy input [49] and even proposes new more general networks, where activation functions can be learned from the data. In addition, analogies between ReLU-based and shrinkage-based [50] models have also been used as an analogy to explain the behavior of CNNs.

Limitations of current theoretical signal processing interpretations of encoding-decoding CNNs. Despite the insights that new theoretical works present to understand the operation of CNNs, the above-described theories also have some limitations. For example, the TDCF focuses mainly on explaining the operation of ReLU-based CNNs. In the case of using sparse coding to understand CNNs, it can be noted that conventional sparse coding behaves in a way that is slightly different than CNNs. For example, sparse coding implementations attempt to find the closest basis functions to the signal after which they produce an approximation of the signal based on those basis functions. The task of finding the closest basis functions to the observed signals is often achieved by means of some optimization mechanism, which is not part of the architecture of CNNs (e.g. k-means [51], the basis-pursuit algorithm [52], or gradient-descent optimization [53]). Finally, in the case of spline functions, it can be observed that the current insights are still limited to ReLU-based approximations and to adaptive piece-wise linear and maxOut functions [49].

Based on the current insights and limitations on the signal processing behavior of ED CNNs, the following section defines the problem statement and research questions for this thesis.

1.5 Problem statement and research questions

This section further details the challenges into a problem statement and specific research questions for this thesis (Sections 1.5.1 and 1.5.2).

1.5.1 Problem statement

Regardless of the significant theoretical developments as well as practical examples which combine signal processing techniques with encoding-decoding CNNs, to the moment of writing this dissertation, most CNN designs do hardly exploit signal processing concepts in their design and are mostly designed empirically. Furthermore, the theory of deep-convolutional framelets and the geometry of encoding-decoding CNNs focuses exclusively on explaining CNNs with ReLU activations. This leads to the following problem statement.

This thesis aims at defining a complementary framework that explains ED CNNs by explicitly connecting architectural elements of the model to existing signal processing functions. The purpose of this explicit connection is to obtain not only a better understanding of CNNs, but also to design new models that leverage the underlying signal processing concepts for achieving a more interpretable operation.

1.5.2 Research questions

The above problem statement results in the following research questions.

RQ 1: Integrating conventional signal processing into CNNs
The theory of deep convolutional framelets acknowledges the convolution filters of the encoder and decoder in noise reduction CNNs as mechanisms for decomposing and reconstructing the input signal, while the sparsifying non-linearities are considered the main noise reduction element. Based on this realization, this thesis proposes two main paths for the design of noise reduction CNNs. The first option is to replace the convolution filters in the encoder and decoder by non-trainable framelets, while the noise reduction relies on CNN models that are applied to specific framelet bands. Alternatively, the second option is to simplify the design of current CNN models to bring them closer to conventional wavelet-based denoising algorithms. Based on the proposed design paths, the following questions are formulated.

- RQ1a. *What are the consequences of the integration of non-trainable framelets as ED path in noise reduction CNNs?*

- RQ1b. *Is it possible to realize a denoising CNN with a trainable ED path that offers perfect reconstruction, without explicitly training for this property?*
- RQ1c. *Can the integration of signal processing knowledge improve CNNs in any aspect?*

RQ 2: Impact of the activation functions in CNNs

The (non-linear) activation functions are one of the key components in the operation of convolutional neural networks. It should be noted that common CNNs use ReLU activations, while the models in this thesis often employ shrinkage activations, which are used also in conventional wavelet-based noise reduction. This modification may impact the behavior of the CNNs, which raises the following questions.

- RQ2a. *Do shrinkage functions have any advantages when compared to ReLUs in terms of CNN interpretability?*
- RQ2b. *How do shrinkage functions limit the operation of CNNs when compared to ReLU activations?*

RQ 3: Comparing characteristics of convolutional neural networks

In the current state of deep learning, it is not obvious how to choose a CNN for a specific application because it is not clear how the deep learning models operate and what limitations they may have. Consequently, it is important to specify a theoretical description or a descriptive framework of the internal operation of a CNN that allows to identify potential limitations. In this way, better-informed choices can be made regarding the suitability of specific CNNs for a given task. By bearing these aspects in mind, the following questions are formulated.

- RQ3a. *Is it possible to define a common framework for the analysis of denoising CNNs to identify their potential limitations?*
- RQ3b. *What are the limitations of such framework?*

RQ 4: Expanding the applications of encoding-decoding CNNs

Model-based deep learning is a technique that at its core uses encoding-decoding CNNs. The model-based deep learning is a solution direction that is applicable to many problems, since it exploits knowledge of the physical process that relates to the analyzed signal. It should be noted that model-based deep learning operates by solving the problem of estimating a desired variable (or a set of variables or data representation) by employing an iterative optimization approach in which a CNN is employed as a regularization step. The concept behind this approach is that by limiting the operation of the CNN to perform regularization, the overall process becomes more explainable. Potentially, this approach is more generic and robust, because model-based deep learning uses the signal model, rather than learning it. This leads to the following questions.

- RQ4a. *Which concepts and principles can be applied to improve the generalization of model-based deep learning?*
- RQ4b. *Can the proposed concepts with ED CNNs also contribute to lower the complexity and how does this compare with conventional CNN designs?*

1.6 Contributions

The scientific contributions of this dissertation are elaborated below.

Contributions to hybrid design of signal processing and CNN systems
The main contribution of this thesis is to present an approach in which signal processing concepts can be integrated in CNNs. The motivation for such integration is to improve the interpretability of the resulting models. In this dissertation, there are two main design paths that are followed.

The first approach employs a framelet decomposition known as the *overcomplete Haar wavelet transform*. This transformation decomposes the input signal in multiple bands. Afterwards, in the high-pass framelet bands, a trainable CNN that is inspired in NeighShrink [54] is employed to eliminate noise. The integration of these elements results in two models referred to as *dual-Haar shrinkage networks* 1 and 2 (DHSN1 and DHSN2, respectively). The conducted experiments show that the DHSN1 and DHSN2 models are competitive in terms of peak signal-to-noise ratio and mean structural similarity index, when compared with other popular denoising CNNs employed in CT. However, the DHSN1 and DHSN2 are more interpretable than common CNN models because the decomposition and reconstruction of the signal is handled by a framelet with known characteristics, while the employed CNNs are limited to eliminate noisy components in the decomposed representation. Furthermore, additional experiments demonstrate that the known sparsity property of the employed framelet can be used for unsupervised noise reduction with the DHSN2.

The second approach for interpretability explores designs where the ED path is learned, but where basic properties of signal preservation can still be observed. Therefore, this approach provides more certainty on the internal operation of the model. To this end, this thesis proposes the (residual) *learned wavelet frame shrinkage network* (LWFSN), which mimics conventional wavelet-based noise reduction, by employing a linear ED path and applying shrinkage non-linearities in the high-pass bands/feature maps. This simplified design allows to study the operation of the model in a better way, since non-linear shrinkage functions become the identity function when the threshold for these activations is set to zero. This concept leverages to study the linear part of the model with the impulse and frequency responses of the system. This demonstrates that the ED path of the proposed LWFSN has the ability to propagate signals almost without any loss, while the non-linearities handle the suppression of noise components. It should be noted that to the best of our knowledge, this research is the first in following this approach to explicitly characterize the behavior of the linear part of the trained model.

Contributions to understanding the activation functions in CNN behavior
Another contribution of this thesis is to demonstrate that shrinkage activations can reduce the model size/complexity of CNNs when compared with ReLU activations. This consequently reduces the execution time when compared to conventional CNNs, as shown in the experiments performed on the *learned wavelet frame shrinkage network*, leading to a over 7 times faster execution, when compared to the tight-frame U-Net

(TF U-Net) [36]. Furthermore, as discussed in the previous contribution, shrinkage activations are beneficial for interpretability of the operation of trained models. In addition, this thesis demonstrates that there are conditions in which ReLUs CNNs and shrinkage functions can be considered equivalent. These conditions are that the number of feature maps is large enough to represent shrinkage activations and that the phase of the filters is suited to correctly propagate the positive and negative parts of the signal. However, if these conditions are satisfied, the bonus is that the use of shrinkage activations always lead to more compact models. For example, the ReLU-based TF U-Net has over 500 times the amount trainable parameters of the shrinkage-based LWFSN.

Contributions to the understanding of ED CNNs

This dissertation exhaustively studies ED CNNs for noise reduction and provides complementary information to the theory of deep convolutional framelets (TDCF) [40]. Specifically, the TDCF explains ReLU CNNs from a low-rank approximation perspective and does not addresses other activations. In addition, an other limitation of this theory is that it has been exploited to find constraints in the propagation of signals in models like the popular U-Net [8]. However, the analysis of the U-Net performed with the TDCF mainly focuses on the linear part of the model, while addressing the ReLUs simply as a "switching" mechanism that reduces the rank of the signal. This means that the implications of the non-linearities in terms of signal propagation/reconstruction has not been explored. In order to account for these limitations, an analysis framework is proposed which considers the implications of the linear and non-linear part of the model in terms of signal reconstruction for ReLU, shrinkage and clipping-based models. Overall, the framework allows to describe networks in such a way that the limitations of CNNs can be found. As an application of the developed framework, this thesis describes the reconstruction and denoising structure of the *residual encoder-decoder CNN* (RED) [37], the *filtered backprojection network* [50] (FBPConvNet), the tight-frame U-Net [36] and the *sparse coding network* [48]. A comparison in Chapter 3 shows the limitations of each of these networks. For example, the analysis performed to the TF U-Net in Chapter 3 shows that the TF U-Net model can be reduced considerably in terms of model complexity. These findings have been employed to design the more compact and faster LWFSN described in Chapter 4, which not only provides a more efficient choice, but also the power of the analysis framework.

Contributions to model-based deep learning

Model-based deep learning integrates regularized optimization with neural networks for more interpretable operation. However, it can be observed that the CNN employed as regularizer still is considered a "black box". The research of this thesis shows that it is possible to incorporate a simpler and more interpretable model as regularizer. Specifically, we employ a model where the encoder and decoder are framelets and in which a simple CNN that eliminates noise is the only trainable part in the model. The integration of the proposed regularizer within the model-based solution allows for noise reduction performance that is close to more conventional model-based CNNs, such as FISTA-Net. However, it should be noted that in terms of PSNR and MSSIM,

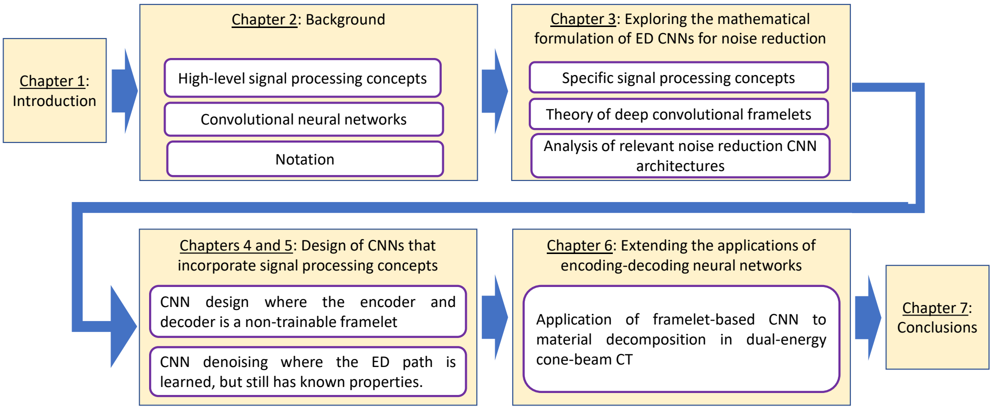

Chapter 1: Introduction
Chapter 2: Background
High-level signal processing concepts
Convolutional neural networks
Notation
Chapter 3: Exploring the mathematical formulation of ED CNNs for noise reduction
Specific signal processing concepts
Theory of deep convolutional framelets
Analysis of relevant noise reduction CNN architectures
Chapters 4 and 5: Design of CNNs that incorporate signal processing concepts
CNN design where the encoder and decoder is a non-trainable framelet
CNN denoising where the ED path is learned, but still has known properties.
Chapter 6: Extending the applications of encoding-decoding neural networks
Application of framelet-based CNN to material decomposition in dual-energy cone-beam CT
Chapter 7: Conclusions

Figure 1.4. Schematic layout of this thesis. Every yellow block in the diagram represents a chapter of this dissertation, whereas the white blocks represent the main contents of the corresponding chapter.

the simplified model is marginally outperformed by the FISTA-Net.

For improving the generalization of the model-based solution explored in Chapter 6, the model-based CNN integrates an adaptive gradient descent solution for optimization, which can be employed in other model-based CNNs. It can be observed that this circumvents the need to learn the gradient steps as occurring in alternative model-based CNNs. Besides this, an other more general concept integrated the proposed design is to make the noise reduction dependent on the noise level of the input, which is inspired by the multi-scale sparse coding network [48]. Both approaches improve the generalization of the proposed solution.

1.7 Thesis outline

This section presents an outline of the chapters in terms of research topics. A schematic diagram of the chapter layout of this text is depicted in Fig. 1.4, where it is visible that the first chapter of this dissertation addresses the introductory content. Meanwhile, Chapter 2 introduces background in noise reduction, CNNs and notation. Chapter 3 explores the mathematical formulation of CNNs from a signal processing perspective. Based on this exploration, Chapters 4 and 5 show CNNs that incorporate framelets in their design to improve their interpretability. As final application, Chapter 6 presents a model-based noise reduction and material decomposition application in the context of dual-energy cone-beam (CB) CT. In this model, the core element is a regularization network which incorporates elements of the designs shown in Chapters 4 and 5. The remainder of this section briefly addresses the contents of each chapter.

Chapter 2 introduces the necessary background on noise reduction, which includes linear filtering, variational methods, transform-domain methods, as well as structured-rank approximation techniques. These concepts are important to understand the developments presented in the succeeding chapters. Furthermore, this chapter also addresses the deep neural networks and encoder-decoder CNNs in more detail. Finally, the last part of Chapter 2 describes the mathematical notation used for the remaining chapters.

Chapter 3 presents an extensive analysis of the connections of encoding-decoding CNNs with signal processing principles, as well as an analysis framework for noise reduction with CNNs. Afterwards, different encoding-decoding CNNs based on ReLU and shrinkage activations are analyzed. This chapter is based on an publication in the *IEEE Signal Processing Magazine* [55].

Chapter 4 addresses the design of a CNN, whose encoding-decoding path is composed by tight wavelet frames. The chapter initially addresses the dual-tree complex wavelet transform (DTCWT), which has certain characteristics that makes it attractive for its integration within a CNN such as filter directionality and perfect reconstruction. Still, the main drawback of the DTCWT is the use of large filters, which reduce their spatial localization and increase the padding needed to avoid signal loss. For a better integration within CNNs, a number of simplifications are introduced to the DTCWT, which results in the so-called *over-complete Haar wavelet transform*. This chapter is based on publications presented at the *IEEE Int. Workshop in Machine Learning for Signal Proc.* (2020) [56] and the journal *IEEE Trans. on Image Proc.* (2021) [57].

Chapter 5 is exploring the design of a CNN in which the encoding and decoding paths are learned, but where it is still possible to prove that from a global point of view, the encoding-decoding path of the model behaves approximately as a tight wavelet frame and its inverse. This property is proved both theoretically and experimentally. The results of this research are published in the journal *IEEE Trans. on Medical Imaging* (2022) [58].

Chapter 6 discusses the insights provided by the designs from the previous two chapters and integrates them into a model-based CNN applied to material decomposition and denoising of dual-energy cone-beam CT scans. The models presented for the experiments in this chapter are trained on a small set of synthetic images and evaluations are conducted in synthetic and clinically-acquired dual-energy cone-beam CT. The results indicate that the proposed model is able to generalize better to unseen images than the reference models. The insights of this research are submitted to the journal *IEEE Trans. on Medical Imaging* [59].

Chapter 7 addresses the final remarks of this dissertation and addresses the research questions formulated at the start of this thesis. In addition, the chapter provides a future outlook for the research presented in this thesis.

CHAPTER 2

Background in noise reduction, sparsity-driven modeling and CNNs

The previous chapter has introduced the context of this thesis, which is the design and understanding of noise reduction CNNs with applications to CT imaging. The development of such CNNs is based on the existing knowledge of diverse existing signal processing algorithms, which rely on assumptions on the signal and noise models. Prior to addressing the design and specific connections that exist between signal processing, noise reduction and CNNs, this chapter summarizes the main principles of relevant signal processing algorithms with the aim of building the foundations for understating their connections with CNNs. Afterwards, the following chapters will use the defined concepts to develop noise reduction algorithms and to draw connections between them and CNNs.

The first concept to be addressed in this chapter is noise reduction, which is an inverse problem where a noiseless estimate is obtained from a noisy observation. Historically, this problem has received significant attention and has been approached from many different perspectives. This dissertation is particularly relevant for the concepts behind sparsity [14], [60], [61], and the structure of the data [13], [62], [63]. It should be noted that under certain considerations some of these techniques can be considered equivalent and/or complementary.

An additional element relevant for this chapter is the concept of neural networks and of encoding-decoding CNNs. It will be shown in Chapter 3 that these concepts have a strong connection with diverse signal processing elements often applied to noise-reduction algorithms. Another important concept introduced in this chapter is the mathematical notation that will be used in the remainder of this text as well as relevant image quality metrics that are used in subsequent chapters.

The remainder of this chapter is structured as follows. Section 2.1 introduces the fundamental aspects of relevant noise reduction algorithms. Afterwards, Section 2.2 introduces the concept of neural networks. Afterwards, Section 2.4 introduces the notation and symbols using in this dissertation, while Section 2.5 presents the image quality metrics that are used in the thesis. Finally, Section 2.6 discusses the conclusions for these sections as well as the challenges to be addressed by the following chapters.

2.1 Existing noise reduction methods

As mentioned in Chapter 1, there are numerous techniques that can be used to reduce noise in images. Among the existing methods, it is possible to find many examples, which are the result of assumptions about the signal and noise characteristics. This section shows a small sample of such techniques, which are also relevant to understand subsequent CNN models in the remainder of this dissertation.

2.1.1 Linear filtering

Linear filtering is perhaps the earliest form of noise reduction. Consequently, there is a wide range noise reduction algorithms based on this principle, which go from simple Gaussian and block filters to model-based algorithms such as Wiener and collaborative filtering. As a first example, convolution-based linear filtering estimates the local mean of the signal with the weighted average/expected value of the signal values around a predefined neighborhood for every pixel. An alternative technique, is the non-causal Wiener filter, where the noisy input is modeled as a noiseless image contaminated with a noise signal that follows a specific distribution. The final step of the Wiener filter is to produce the least-squares estimate of the signal, which is often performed in the Fourier domain. The final technique of linear noise reduction addressed here is non-local or collaborative filtering, which appeals to the redundancy of natural images by grouping sets of similar patches and performing a weighted average. The most common of these algorithms is non-local means [64].

2.1.2 Variational methods

Variational methods are optimization-driven techniques, where noise reduction is achieved by minimizing an objective function which enforces desirable properties in the resulting noiseless estimate. The most common example of a variational noise reduction technique is the model by Rudin Osher and Fatemi (ROF) [13], which is mathematically expressed by

$$\hat{\mathbf{y}} = \underset{\mathbf{z}}{\operatorname{argmin}} \left[F(\mathbf{x}, \mathbf{z}) + R(\mathbf{z}) \right], \tag{2.1}$$

where variable $\hat{\mathbf{y}}$ is an estimate of a noiseless image $\mathbf{y}$, while $\mathbf{x}$ is the observed noise-contaminated image and $\mathbf{z}$ is an intermediate representation of the noiseless image $\mathbf{y}$. In addition, function $F(\cdot)$ is the *fidelity* term, which enforces the estimate $\mathbf{z}$ to remain close to the noisy input $\mathbf{x}$, while the term $R(\cdot)$ is a *regularization* term, which introduces prior information about the image estimate. For example, natural images are often considered to be locally smooth/constant. Therefore, the $R(\cdot)$ could penalize non-smooth images.

In the popular ROF method, functions $R(\cdot)$ and $F(\cdot)$ are defined by $R(\mathbf{z}) = \|\mathbf{z}\|_{\mathrm{TV}}$ and $F(\mathbf{x}, \mathbf{z}) = \lambda \cdot \|\mathbf{x} - \mathbf{z}\|_2^2$, respectively. Here, $\|\cdot\|_{\mathrm{TV}}$ stands for the *total variation* norm, while $\|\cdot\|_2$ is the L^2 norm and constant λ defines a trade-off between fidelity with

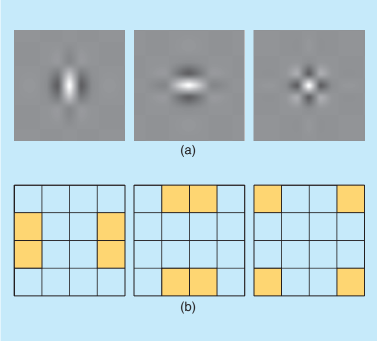


Figure 2.1. Example of detail/high-pass filters of the discrete wavelet transform (subfigure (a)) and their frequency response (subfigure (b)). Image extracted from the work by Selesnick and Kingsbury [65] © 2005 copyright IEEE.

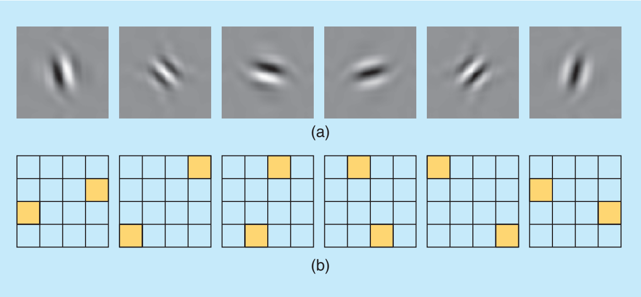


Figure 2.2. Real part of the filters of the detail filters of the dual-tree complex wavelet transform (subfigure (a)) and their associated frequency spectrum (subfigure (b)). Image extracted from the work by Selesnick and Kingsbury [65] © 2005 copyright IEEE.

respect to the input signal and smoothness. The total variation norm $\|\cdot\|_{\mathrm{TV}}$ is defined by

$$\|\mathbf{x}\|_{\mathrm{TV}} = \|\nabla\mathbf{x}\|_1, \tag{2.2}$$

in which $\nabla\cdot$ is the divergence operator that is often implemented with a first one-sided finite difference.

In order to solve the optimization problem of Eq. (2.1), multiple solutions have been proposed, such as finite difference solvers [13], primal-dual optimization [66], Brengman iterations [67]–[69] and fixed-point methods [70]. All these optimization procedures result in iterative solutions.

2.1.3 Transform-domain methods

Transform-based denoising leverages the idea that an image can be mapped to a sparse domain, in which most of its energy is contained within a few high-amplitude components, whereas noise is distributed in low-amplitude samples. It should be noted that this separation is useful and can be exploited for noise removal. Examples

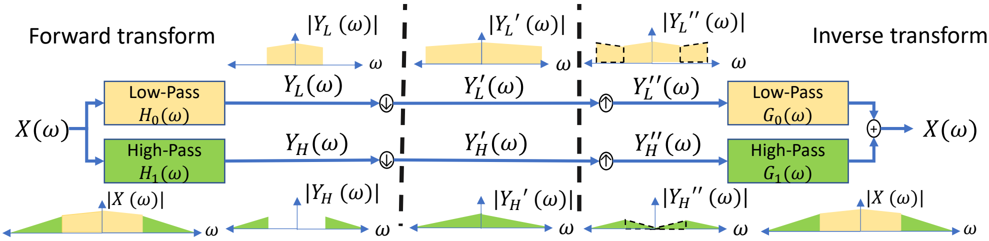


Figure 2.3. Processes involved in the computation of the forward and inverse wavelet transform and their effect in the Fourier spectrum of the input signal. In the figure, $X(\omega)$ is the input signal in the Fourier domain, H_0 and H_1 are low-pass and high-pass filters used in the forward transform, respectively, while G_0 and G_1 are their inverse transform counterparts. After the first processing stage, the signal $X(\omega)$ is bisected into the bands $Y_L(\omega)$ and $Y_H(\omega)$, since only half of the bandwidth is used. Then, $Y_L(\omega)$ and $Y_H(\omega)$ are down-sampled, resulting in signals $Y_L'(\omega)$ and $Y_H'(\omega)$. Finally, during the inverse transformation, the encoded signals $Y_L'(\omega)$ and $Y_H'(\omega)$ are up-sampled, resulting in the aliased signals $Y_L''(\omega)$ and $Y_H''(\omega)$, which can be anti-alias filtered by G_0 and G_1 and later added to recover recover the signal $X(\omega)$. It should be noticed that the reconstructed signal has no prime indication because of the perfect reconstruction.

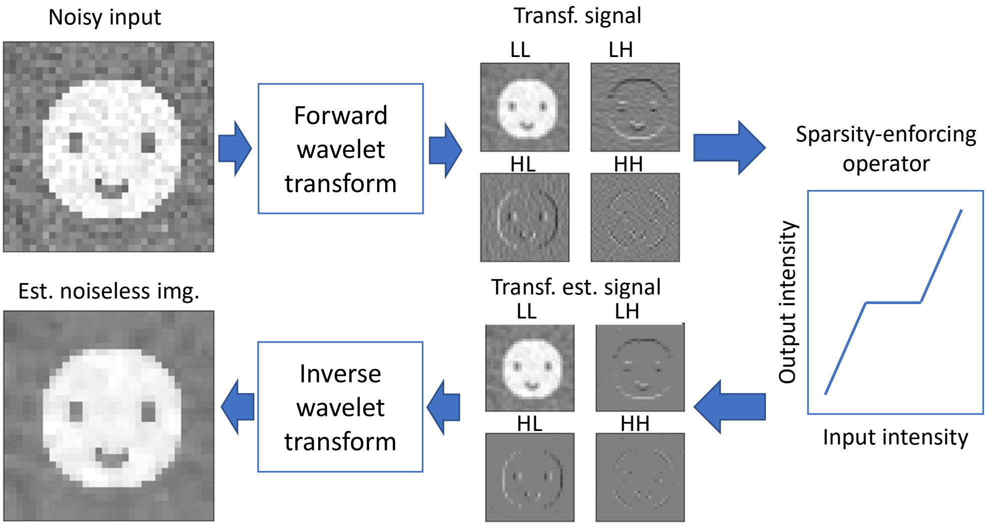


Figure 2.4. Simple wavelet-based denoising pipeline. In the first place, the signal is converted to the wavelet domain. The second step consists in applying a non-linear sparsity-enforcing operator. Finally, the signal is mapped back to the original domain with the inverse wavelet transform.

of such transforms are the Discrete Cosine Transform [71], orthogonal wavelets [14], [54], [60], [61], [72] and redundant transformations such as framelets/shearlets [73]–[75].

The discrete wavelet/framelet transform is a signal representation that maps a (single-) channel signal to a multi-channel domain, where each channel contains a fraction of the (Fourier) spectrum of the original image. The mapping to the wavelet domain is achieved by convolving the input image with filters referred to as wavelet *bases/kernels* (e.g. see Fig 2.1). An advantage of the multi-channel representation of wavelets/framelets is that every channel contains only a fraction of the total bandwidth of the original signal. Furthermore, the bandwidth-limited representation allows to down-sample the resulting channels with minimal aliasing. This property enables multi-scale processing by recursively decomposing the low-frequency band/channel of the previous decomposition level. A one-dimensional simplified example of the spectrum partitioning of the wavelet decomposition is depicted in Fig. 2.3.

Although wavelets/framelets yield good space/frequency partitioning characteristics, using the orthogonal DWT, it has some limitations. For example, filters of an orthogonal DWT are shown in Fig. 2.1, where it is visible that the directionality of the conventional DWT is limited (i.e. the filters are aligned only with directions $0°$, $90°$ and $\pm 45°$). The directionality issues of the DWT can be solved with the use of redundant framelet decompositions, such as the dual-tree complex wavelet transform (DTCWT) [74]. For reference, part of the filters of the DTCWT are displayed in Fig. 2.2. It can be observed that the filters of the DTCWT divide the Fourier spectrum of the signal into narrower sections than the DWT, which results in better directionality and less aliasing than the conventional DWT.

Finally, it should be considered that there are many methods for denoising in the framelet/wavelet domain. However, most of them follow the procedure displayed in Fig. 2.4, based on the following steps. First, a noisy input is converted to the framelet/wavelet domain. Second, a non-linear sparsity-enforcing function is applied. Third, the noiseless estimate is mapped back to the original domain. It should be noted that the thresholding procedure is often derived from statistical properties of the image and noise [14], [76].

2.1.4 Rank-based methods

An alternative formulation for modeling signals is to assume that every image patch is generated by the superposition of simpler image bases. This concept has been applied by denoising algorithms, such as sparse dictionaries [15] and (structured) low-rank approximation [62], [63], [78]. Moreover, there are also methods like the low-rank approximation based on Hankel-structured matrices, in which every image patch is lifted to a matrix Hankel representation. This last approach has been successfully used for removal of impulse noise [77], [79], [80] and an example of such algorithm is the robust *annihilating-filter-based low-rank Hankel matrix approach* (ALOHA) [77], which is described in more detail below.

The global operation of ALOHA is shown in Fig. 2.5, while its signal model is depicted in Fig. 2.6. In addition, the steps of robust ALOHA based on the referred

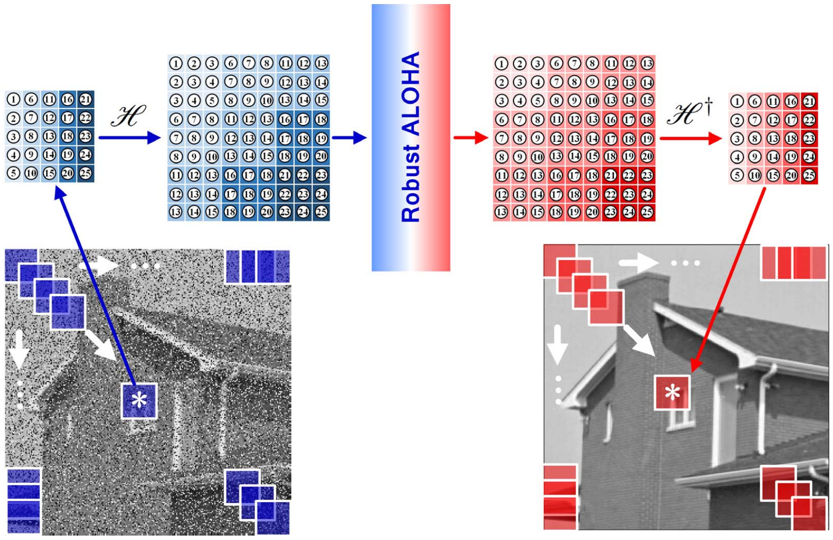


Figure 2.5. High-level description of the robust *annihilating-filter-based low-rank Hankel matrix approach* (ALOHA). This algorithm is designed to remove impulse noise from images and it is composed of the following steps. (1) A sliding window is applied to extract image patches. (2) The extracted patches are lifted to a Hankel-matrix representation with operator $\mathscr{H}$. (3) The resulting Hankel-structured matrix is denoised with the robust ALOHA algorithm. (4) The estimated noiseless Hankel-structured matrix is mapped back to an image patch. (5) The final step is to aggregate all the estimated patches in the final noiseless image estimate. Image from Jin *et al.* [77] © 2017 copyright IEEE.

diagrams are described as follows.

1. Image patches are extracted with overlap (represented in Fig. 2.6 by signal $M[i, j]$).
2. The extracted patches are lifted to a Hankel matrix representation (in Fig. 2.6 the lifted patch is the signal $L + S$, where L is the low-rank Hankel matrix and S represents the sparse outliers).
3. The Hankel representation is modeled as the superposition of sparse and structured outliers S plus a low-rank (noiseless) signal L and an optimization approach based on annihilating filters is applied, to produce an estimate $\hat{L}$ of the low-rank component L.
4. The estimated low-rank Hankel matrix $\hat{L}$ is mapped back to an image patch $\hat{M}[i, j]$ and spatially averaged with surrounding estimates. This procedure generates the final estimated noiseless picture.

It should be noted that robust ALOHA has been referred to as the main inspiration for the theory of deep convolutional framelets [40], which explains the operation of denoising ReLU CNNs and which is later addressed in Chapter 3.

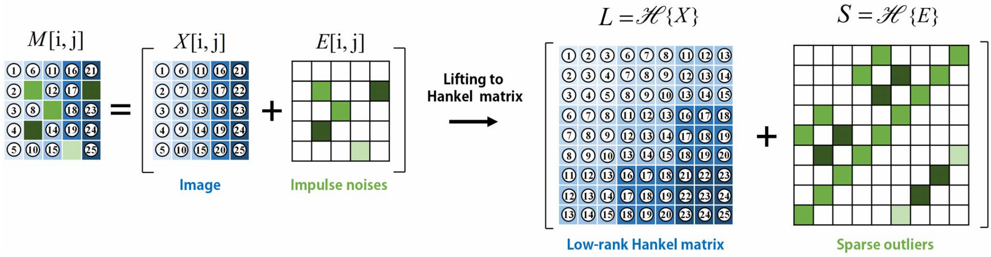


Figure 2.6. Signal model of the ALOHA algorithm. The robust ALOHA algorithm assumes that the observed patch $M[i, j]$ (where $[i, j]$ denotes the patch location) is composed by the superposition of a noiseless image $X[i, j]$ and a noise patch $E[i, j]$. As second step, the signal $M[i, j]$ is lifted to a Hankel representation (denoted by the operator $\mathscr{H}(\cdot)$). This approach results in a patch $P = L + E$, where $L = \mathscr{H}(X)$ is the low-rank component that represents the image, while $S = \mathscr{H}(E)$ is the Hankel representation of the impulse noise. Image from Jin *et al.* [77] © 2017 copyright IEEE.

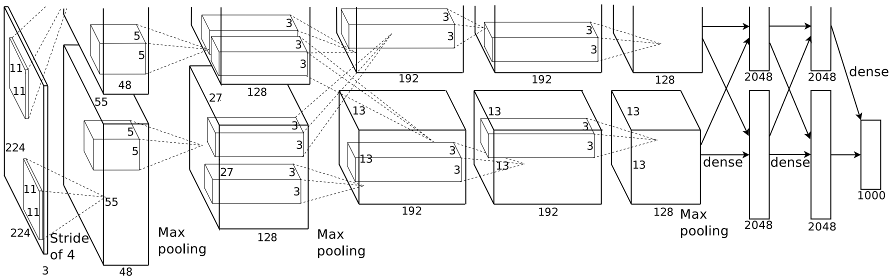


Figure 2.7. Diagram of the AlexNet architecture. This is is one of the first CNNs with integrated multiple convolution layers, using ReLU activations, max-pooling and fully-connected layers in the final classification stage. Image from Krizhevsky *et al.* [5].

2.2 Convolutional neural networks

Over the past decade, convolutional neural networks have achieved state-of-the-art performance and become the *de-facto* solution for diverse computer vision tasks, such as image classification [5], [81], [82], image detection [7], [83], [84], image segmentation [8], [85], and image denoising [35].

AlexNet [5] is a CNN used for classifying images that established the deep learning breakthrough because it outperforms feature-based classifiers. Highlights of the AlexNet architecture with respect preceding models are the use of rectified linear units (ReLUs) [86], as well as of multiple sequential convolution and down-sampling layers. The AlexNet model is displayed in Fig. 2.7 and the filters learned by this model are shown in Fig. 2.8. It should be noted that the filters learned by this CNN are reminiscent of the framelet/wavelet kernels discussed earlier (i.e. see Figs. 2.2 and 2.8). The commonalities between the learned filters by CNNs and framelets will be addressed later in this thesis.

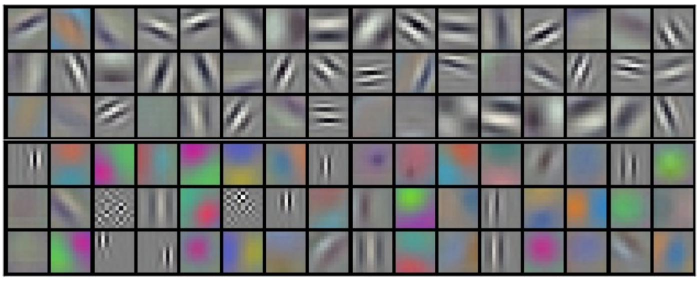


Figure 2.8. Filters learned by one of the layers of the AlexNet architecture. It can be observed that these filters are similar to the basis functions of the DTCWT depicted in Fig. 2.2. Image from Krizhevsky *et al.* [5].

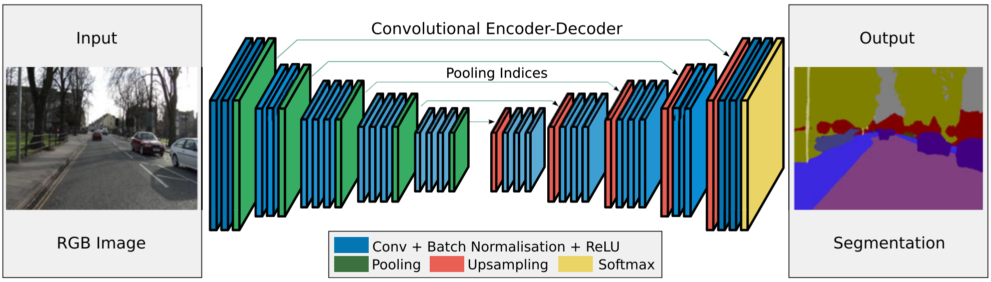


Figure 2.9. Diagram of SegNet CNN. This model is an early example of a fully convolutional encoding-decoding CNN. Image from Badrinarayanan *et al.* [85]

Since the early AlexNet model, CNNs have evolved and have permeated into many applications. The rest of this section focuses on the *encoding-decoding* CNNs, which are used in many pixel-level tasks [8], [85], [87] and are the mostly used concept of network modeling in this dissertation.

2.3 Encoder-decoder architectures

The origins of encoding-decoding CNNs are rooted in techniques for dimensionality reduction and unsupervised feature extraction. Specifically, one of the oldest encoder-decoder architectures is *principal component analysis* (PCA), where a set of high-dimensional feature vectors are mapped into a low-dimensional space with a linear projection (*encoder*). The processed lower-dimensional data point is known as *code*. Finally, the code could be mapped back to the original data dimensionality with another linear projection (*decoder*).

In the specific context of neural networks, an early encoder-decoder (or autoencoder) example is Kramer's model [88], which reduces the dimensionality of the codes produced by PCA, by employing a single-layer encoding-decoding pair with sigmoid activations. Alternatives to Kramer's model are performed by Rumelhart, Hinton and Williams [89], who have introduced the back-propagation and initialization algorithms for training multiple encoding-decoding layers. Later, more approaches have been developed to further enhance the representations learned by autoencoder models. For example, Vincent *et al.* [90] proposed the concept of a denoising autoencoder, where the autoencoder is trained to produce noise-free estimates from noise-contaminated inputs.

The preceding text has introduced only autoencoders that perform dimensionality reduction. However, it should be noticed that besides the low-dimensionality property, other features may be desirable. One the these properties is *sparsity*, which indicates when most of the energy of a signal is concentrated in a few parameters. An example of an autoencoder enforcing sparsity is the model by Alireza and Brendan [91], whose training incorporates constraints that increase the sparsity of the generated codes. An alternative conception of the autoencoder is to use convolution filters, instead of fully-connected layers. This results in designs such as the model by Ranzato *et al.* [92], which is one of the first encoding-decoding *convolutional* neural networks (CNNs). For the remainder of this dissertation, any reference to encoding-decoding CNNs follows the same structure of Ranzato's design.

After the wide adoption of CNNs for image classification, these models have been introduced to pixel-level tasks such as segmentation. These early attempts use CNNs to label every pixel in the image and often refine this result with approaches such as conditional random fields [93], [94]. It should be noted that the pixel-level classification approach is slow, because it requires thousands (or even millions) of evaluations to process a full image. The disadvantages of pixel-level classifiers for image segmentation are circumvented by works such as from Ronnenberger *et al.* [8], Badrinarayanan *et al.* [85] (see Fig. 2.9) and Miletari *et al.* [87], all of which propose segmentation models based on fully convolutional neural networks that perform end-to-end image segmentation. Furthermore, the referred models have encoding-decoding convolutional structures, akin to the model by Ranzato *et al.* [92]. Given the success of CNNs in image segmentation, the fully convolutional encoding-decoding CNN concept has been quickly adopted for multiple image-to-image tasks, such as image denoising.

2.4 Notation

Convolutional neural networks (CNNs) are composed by basic functions such as convolutions, activations and down/up-sampling. In order to achieve better clarity in the explanations given in this dissertation, this section defines the mathematical notations being used to represent the internal operations of CNNs.

In this thesis, a scalar is represented by a lower-case letter (for example a), while a vector is represented by an underlined lower-case letter (e.g. $\underline{b}$). Furthermore, a matrix –such as an image or convolution mask– is represented by a boldface letter (e.g. variables $\mathbf{x}$ and $\mathbf{y}$). Finally, a tensor is defined by an underlined uppercase letter in boldface. For example, the two arbitrary tensors $\underline{\mathbf{A}}$ and $\underline{\mathbf{Q}}$ are defined by

$$\underline{\mathbf{A}} = \begin{pmatrix} \mathbf{a}_0^0 & \dots & \mathbf{a}_{N_\mathrm{C}-1}^0 \\ \vdots & \ddots & \vdots \\ \mathbf{a}_0^{N_\mathrm{R}-1} & \dots & \mathbf{a}_{N_\mathrm{C}-1}^{N_\mathrm{R}-1} \end{pmatrix}, \underline{\mathbf{Q}} = \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}^{N_\mathrm{R}-1} \end{pmatrix}. \tag{2.3}$$

Here, entries $\mathbf{a}_c^r$ and $\mathbf{q}^r$ represent two-dimensional arrays (matrices). Since the defined tensors are used in the context of CNNs, matrices $\mathbf{a}_c^r$ and $\mathbf{q}_c$ can be learned filters, which have dimensions $(N_\mathrm{H} \times N_\mathrm{V})$, where N_H and N_V denote the filter dimensions in the horizontal and vertical directions, respectively. Finally, the total tensor dimension of $\underline{\mathbf{A}}$ and $\underline{\mathbf{Q}}$ are defined by $(N_\mathrm{R} \times N_\mathrm{C} \times N_\mathrm{H} \times N_\mathrm{V})$ and $(N_\mathrm{R} \times 1 \times N_\mathrm{H} \times N_\mathrm{V})$, where N_R and N_C are the number of row and column entries, respectively. If the tensor $\underline{\mathbf{A}}$ contains the convolution weights in a CNN, the row-entry dimensions represent the input number of channels to a layer, while the number of column elements denotes the number of output channels.

Having defined the notation for the variables, a few relevant operators are discussed. First, the transpose $(\cdot)^\intercal$ of the previously defined tensor $\underline{\mathbf{Q}}$ (here expressed by $\underline{\mathbf{Q}}^\intercal$) switches its row and column entries. This is shown by the mathematical expression

$$\underline{\mathbf{Q}}^\intercal = \begin{pmatrix} \mathbf{q}_0 & \dots & \mathbf{q}_{N_\mathrm{R}-1} \end{pmatrix}. \tag{2.4}$$

Note that the transpose of the tensor is different from the transposed convolution often used in CNNs, since the transposed convolution is an up-sampling layer. Furthermore, the tensor convolution $\underline{\mathbf{A}} \circledast \underline{\mathbf{Q}}$ is specified by

$$\underline{\mathbf{A}} \circledast \underline{\mathbf{Q}} = \begin{pmatrix} \sum_{r=0}^{R-1} \mathbf{a}_r^0 * \mathbf{q}_r \\ \vdots \\ \sum_{r=0}^{R-1} \mathbf{a}_r^{N_\mathrm{C}-1} * \mathbf{q}_r \end{pmatrix}. \tag{2.5}$$

Here, operator $\circledast$ is the tensor convolution, while symbol $*$ defines the convolution between two matrices (images).

In this dissertation, images –which are 2D arrays (matrices)– are often convolved with 4D tensors. When this operation is performed, images are considered to have dimensions $(1 \times 1 \times N_\mathrm{H} \times N_\mathrm{V})$, where N_H and N_V denote the number of elements in the horizontal and vertical directions, respectively. In addition, in this dissertation matrix $\mathbf{I}$ is the identity signal for the convolution operator, which for a two-dimensional image is the Kronecker delta/discrete impulse (an image with a single non-zero pixel in the middle with unity amplitude). In order to keep the notation simple, the same symbol for identity will be used regardless of the number of channels of the signal. Furthermore, the variables in the *decoding path* of a CNN are distinguished with a tilde (e.g. $\underline{\tilde{\mathbf{K}}}$, $\underline{\tilde{b}}$). This notation applies to trainable and non-trainable elements.

Additional symbols that are used throughout this dissertation are the down-sampling and up-sampling operations by a factor s, which are denoted by $f_{(s\downarrow)}(\cdot)$ for down-sampling and $f_{(s\uparrow)}(\cdot)$ for up-sampling. Here, both operations are defined in the same way as in multi-rate filter banks [42]. Furthermore, other operators such as

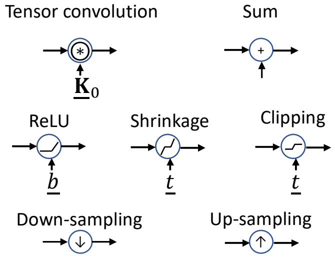


Figure 2.10. Symbols used for the schematic representations of CNNs. It can be observed that the down-sampling and the up-sampling layers do not specify up/down-sampling factor s, which is typically 2 in this thesis.

rectified linear unit (ReLU) bye $(\cdot)_+$, the *shrinkage/thresholding* by $\tau_{(\cdot)}(\cdot)$ and the *clipping* activations by $\mathcal{C}_{(\cdot)}(\cdot)$.

For better clarity, the most important symbols used in this thesis are summarized in Table 2.1. In addition, their graphical representations are shown in Fig. 2.10.

2.5 Image quality metrics

This section summarizes the main image similarity metrics that are employed to assess the denoising performance of the analyzed models in this thesis. The first image-quality metric described here is the signal-to-noise ratio (SNR), which measures the relative power of the noiseless and noise images. This metric is mathematically described by

$$\mathrm{SNR}(\mathbf{x},\mathbf{y}) = 10\cdot\log_{10}\left(\frac{\sum_{h=0}^{N_\mathrm{H}-1}\sum_{v=0}^{N_\mathrm{V}-1}(\mathbf{x}[h,v])^2}{\sum_{h=0}^{N_\mathrm{H}-1}\sum_{v=0}^{N_\mathrm{V}-1}(\mathbf{x}[h,v]-\mathbf{y}[h,v])^2}\right), \tag{2.6}$$

in which $\mathbf{x}$ is the noiseless image and $\mathbf{y}$ indicates the noise-contaminated version of $\mathbf{x}$, while indices h and v are parameters for the horizontal and vertical directions, respectively. In addition, N_H and N_V are the number of pixels in the horizontal and vertical dimensions, respectively. Complementary to the SNR, this dissertation employs also the peak signal-to-noise ratio (PSNR), described by the expression

$$\mathrm{PSNR}(\mathbf{x},\mathbf{y}) = 10\cdot\log_{10}\left(\frac{\mathrm{Max}^2}{\mathrm{MSE}(\mathbf{x},\mathbf{y})}\right). \tag{2.7}$$

Here, $\mathrm{MSE}(\mathbf{x},\mathbf{y})$ is the mean squared error between the images $\mathbf{x}$ and $\mathbf{y}$, while Max is the maximum value that can be contained in the images $\mathbf{x}$ and $\mathbf{y}$.

The (P)SNR does not fully reflect the local similarity between two images. In order to address this limitation, Wang *et al.* [95] propose the structural similarity index metric (SSIM), which is defined by

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\boldsymbol{\mu}_\mathbf{x}\boldsymbol{\mu}_\mathbf{y}+c_1)(2\boldsymbol{\sigma}_{\mathbf{xy}}+c_2)}{(\boldsymbol{\mu}_\mathbf{x}^2+\boldsymbol{\mu}_\mathbf{y}^2+c_1)(\boldsymbol{\sigma}_\mathbf{x}+\boldsymbol{\sigma}_\mathbf{y}+c_2)} \tag{2.8}$$

Here, $\boldsymbol{\mu}_{\mathbf{x}}$ and $\boldsymbol{\mu}_{\mathbf{y}}$ represent the pixel means for images $\mathbf{x}$ and $\mathbf{y}$, respectively. Likewise, images $\boldsymbol{\sigma}_{\mathbf{x}}$ and $\boldsymbol{\sigma}_{\mathbf{y}}$ are the pixel variances of images $\mathbf{x}$ and $\mathbf{y}$, while $\boldsymbol{\sigma}_{\mathbf{xy}}$ is the covariance between them. Moreover, constants c_1 and c_2 are employed for numerical stability. For the experiments here presented, the mean of the SSIM (MSSIM) is employed as image-quality metric and its implementation is found in the computational package Scipy [96].

Despite the fact that the (M)SSIM accounts for local similarity between two images, this metric does not always correlate with the *perceptual* similarity between them. This limitation has been addressed by the *Haar-wavelet-based perceptual similarity index* (HaarPSI) [97]. The referred index uses Haar wavelets as prepossessing to compute the local similarity, as well as a weighting function based on the concept of *phase congruency* [98]. For the computation of this metric, this dissertation uses the routine provided by Reisenhofer *et al.* [97].

2.6 Conclusions

This chapter has presented introductory materials in noise reduction and neural networks, which are key concepts for understanding the discussions addressed in this dissertation. Moreover, this chapter has introduced the mathematical notations being used to describe CNNs and an overview of the most important quality metrics that are used to compare image estimates. The next chapter elaborates on the commonalities between signal processing algorithms and encoding-decoding (ED) CNNs.

Symbol	Meaning
$G(\cdot)$	Generic encoding-decoding CNN.
$\mathbf{I}$	Convolution identity.
$\underline{\mathbf{K}}$	Encoding convolution kernel.
$\underline{\tilde{\mathbf{K}}}$	Decoding convolution kernel.
$\underline{\mathbf{W}}$	Filters for the forward discrete wavelet transform.
$\underline{\tilde{\mathbf{W}}}$	Filters for the inverse discrete wavelet transform.
$\underline{\mathbf{W}}_{\mathrm{H}}$	High-pass filters of the forward discrete wavelet transform.
$\underline{\mathbf{W}}_{\mathrm{L}}$	Low-pass filter of the forward discrete wavelet transform.
$\mathbf{x}$	Noiseless image.
$\mathbf{y}$	Noisy image.
$\boldsymbol{\eta}$	Additive noise.
$\underline{b}$	Bias vector.
t	Threshold level.
$f_{(2\downarrow)}(\cdot)$	Down-sampling operation.
$f_{(2\uparrow)}(\cdot)$	Up-sampling operation.
$*$	Image convolution.
$\circledast$	Tensor convolution.
$(\cdot)^{\intercal}$	Transpose of a tensor.
$(\cdot)_{+}$	ReLU activation.
$\tau_{(\cdot)}(\cdot)$	Generic thresholding/shrinkage operation.
$\mathcal{C}_{(\cdot)}(\cdot)$	Generic clipping operation.

Table 2.1. Mathematical notation of the most common symbols and variables used in this thesis.

CHAPTER 3

Encoding-decoding CNNs for image denoising

3.1 Introduction

The previous chapter shows noise reduction algorithms based on signal processing, which employ direct assumptions on the signal and noise characteristics, such as wavelet shrinkage [14], [60] and low-rank approximation methods [99]. With the advent of deep learning techniques, signal processing algorithms for image denoising have been regularly outperformed and increasingly replaced by encoding-decoding convolutional neural networks (CNNs).

In contrast to conventional signal processing algorithms, encoding-decoding CNNs are often presented as a solution that does not make explicit assumptions on the signal and noise. For example, in supervised CNN-based algorithms, an encoding-decoding CNN *learns* the optimal parameters to filter the signal from a set of paired examples of noise/artifact-free images and images contaminated with noise/artifacts [9], [36], [37]. This approach highly simplifies the solution of the noise reduction problems, since this circumvents the use of explicit modeling of the signal and noise. Furthermore, the good performance and simple use of encoder-decoder CNNs/autoencoders have enabled additional data-driven noise reduction algorithms, where CNNs are embedded as part of a larger system. Examples of such approaches are unsupervised noise reduction [100] and denoising based on generative adversarial networks [101]. Besides this, smoothness in signals can be obtained also by advanced regularization using CNNs, e.g. by exploiting data-driven model-based iterative reconstruction [102], [103].

Challenges. Despite of the flexibility and good performance of CNNs for noise reduction, the high-complexity of these models in terms of number of parameters and non-linear behavior pose a challenge for the interpretability of CNNs. This is a major downside for medical applications, because the black-box approach can hide potential failure cases of these models where critical content of medical images is compromised. Consequently, for the professional medical area, more transparent models where the treatment applied to the signal is well visible and understood, is highly desirable. Hence, the good noise reduction performance of auto-encoding models is attractive, but these models have also the following challenges that should be considered.

- *Explainability*. The complexity and heuristic nature of encoding-decoding CNNs often offers restricted understanding of the internal operation of such architectures [17].

- *Computational load*. The training and deployment of CNNs requires specialized hardware and the use of significant computational resources. Therefore, attention has to be paid to the efficiency of algorithms and the design of the model.

- *Unclear model limitations*. The restricted understanding of the signal modeling in encoding-decoding CNNs does not clearly reveal the limitations of specific models and, consequently, it is not evident how to overcome these problems.

Background. In order to overcome the limitations of encoding-decoding CNNs, new research has addressed the lack of explainability of these models by acknowledging the similarity of the building blocks of encoding-decoding CNNs and the elements of well-known signal processing algorithms, such as wavelet decomposition, low-rank approximation [40], [43], [104], variational methods [105], lower-dimensional manifolds [100] and sparse coding [41]. Furthermore, practical works based on shrinkage-based CNNs inspired by well-established wavelet shrinkage algorithms, has further deepened the connections between signal processing and CNNs [39], [48]. This unified treatment of signal processing-inspired CNNs has resulted in more explainable designs [36], [100], better model performance [36] and improved memory efficiency [58].

Direction of the approach. This chapter has three main objectives. First, to summarize the diverse explanations of the components of encoding-decoding convolutional neural networks applied to image noise reduction, based on the concept of deep convolutional framelets [40] and on elementary signal processing concepts. Both aspects are considered with the aim of achieving an in-depth understanding of the internal operation of encoding-decoding CNNs and to show that the design choices make *implicit* assumptions about the signal characteristics inside the CNN. A second objective is to offer practitioners tools for optimizing their CNN designs with signal processing concepts. Third and final, this chapter presents practical use cases, where existing CNNs are analyzed in a unified framework, thereby enabling a better comparison of different designs, by making their internal operation explicitly visible and paving the way for introducing the proposed designs from Chapters 4 and 5 .

Layout. The structure of this chapter is as follows. Section 3.2 describes the signal model and the architecture of encoding-decoding networks. Afterwards, Section 3.3 addresses fundamental aspects of signal processing, such as singular value decomposition, low-rank approximation, framelets, as well as the estimation of signals in the framelet domain. All the concepts of Sections 3.2 and 3.3 converge in Section 3.4, where the encoding-decoding CNNs are interpreted in terms of a data-driven low-rank approximation and of wavelet shrinkage. Afterwards, based on the learnings from Section 3.4, Section 3.5 shows the analysis of diverse architectures from a signal processing perspective and under a set of explicit assumptions. Finally, Section 3.8 elaborates concluding remarks.

3.2 Encoding-decoding CNNs

This section aims to define the signal model and a high-level description of an encoding-decoding CNN. Specifically, Subsection 3.2.1 introduces the additive-noise signal model. Meanwhile, Section 3.2.2 addresses the elementary parts of an encoding-decoding CNN. These definitions will be used in subsequent sections to connect them with signal processing concepts.

3.2.1 Signal model and noise reduction configurations

In noise reduction applications, the common additive signal model is defined by

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\eta}, \tag{3.1}$$

where the observed signal $\mathbf{y}$ is the result of contaminating a noiseless image $\mathbf{x}$ with additive noise $\boldsymbol{\eta}$. For many applications the noiseless signal $\mathbf{x}$ is estimated from the noisy observation $\mathbf{y}$. In deep learning applications, this is often achieved by models with the form

$$\hat{\mathbf{x}} = G(\mathbf{y}). \tag{3.2}$$

Here, $G(\cdot)$ is a generic encoding-decoding CNN. We refer to this form of noise reduction as *non-residual*, where the model approaches to estimate $\mathbf{x}$ directly from $\mathbf{y}$ as good as possible. Alternatively, it is possible to find $\hat{\mathbf{x}}$ by training $G(\cdot)$ to estimate the noise component $\hat{\boldsymbol{\eta}}$, and subtract it from the noisy image $\mathbf{y}$ to estimate the noiseless image $\hat{\mathbf{x}}$, or equivalently

$$\hat{\mathbf{x}} = \mathbf{y} - G(\mathbf{y}). \tag{3.3}$$

This model is referred to as *residual* [9], [37], [50], because the output of the network is subtracted from its input. For reference, Fig. 3.1 portrays the difference of the placement of the encoding-decoding structure in residual and non-residual configurations.

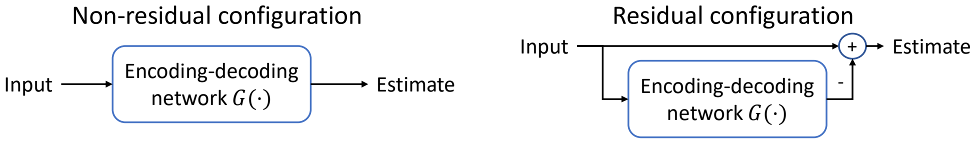

Non-residual configuration
Input
Encoding-decoding network $G(\cdot)$
Estimate
Residual configuration
Input
+
Estimate
-
Encoding-decoding network $G(\cdot)$

Figure 3.1. Residual and **non-residual** network configurations for the encoding-decoding model. Note that the main difference between both designs is the global skip connection occurring in the residual structure.

3.2.2 Parts of encoding-decoding CNNs

It can be observed that encoding-decoding CNNs are constituted of three main parts. (1) The **encoder**, which maps the incoming image to a more redundant representation (i.e. the number of channels of the tensor increases), using a convolution layer. Every channel of the resulting redundant representation contains a fraction of the content of the original signal. It should be noted that the encoder often (but not necessarily) decreases the resolution of the high-dimensional representation, to enable multi-resolution processing and to decrease the memory requirements of the design. (2) The **non-linearities**, which suppress specific parts of the signal. (3) The **decoder**, which maps the multi-channel representation back to the original space. In summary, the most basic encoding-decoding step in a CNN $G(\mathbf{x})$ is expressed by

$$\boxed{G(\mathbf{y}) = G_{\text{dec}}(G_{\text{enc}}(\mathbf{y}))}\ , \tag{3.4}$$

where $G_{\text{enc}}(\cdot)$ is the encoder, which is generally defined by

$$\begin{aligned} \underline{\mathbf{C}}_0 = E_0(\mathbf{y}), \quad \underline{\mathbf{C}}_1 =& E_1(\underline{\mathbf{C}}_0), \quad \underline{\mathbf{C}}_2 = E_2(\underline{\mathbf{C}}_1), \\ &\vdots \\ \underline{\mathbf{C}}_{N-1} =& E_{N-1}(\underline{\mathbf{C}}_{N-2}), \\ G_{\text{enc}}(\mathbf{y}) =& \underline{\mathbf{C}}_{N-1}. \end{aligned} \tag{3.5}$$

Here, $\underline{\mathbf{C}}_n$ represents the encoding generated by the n-th encoder $E_n(\cdot)$. It should be noted that this chapter makes extensive use of operators such as convolutions, up/down-sampling layers and variables with diverse dimensions such as images, vectors and tensors. The notation to distinguish these elements is defined in detail in Section 2.4 and adopted in this chapter. From Eq. (3.5), the encoder layers $E_n(\cdot)$ are defined by the equation

$$\underline{\mathbf{C}}_n = E_n(\underline{\mathbf{C}}_{n-1}) = f_{(s\downarrow)}(A_{(\underline{b}_{n-1})}(\underline{\mathbf{C}}_{n-1} \circledast \underline{\mathbf{K}}_{n-1})). \tag{3.6}$$

Here, the function $A(\cdot)_{(\cdot)}$ is a generic activation used in the encoder, $f_{(s\downarrow)}(\cdot)$ is a down-sampling function by factor s and $\underline{\mathbf{K}}_n$ and $\underline{b}_n$ are the convolution kernel and biases of the *n-th* encoding layer. Complementary to the encoder, the decoder network maps the multi-channel sparse signal back to the original domain. Here, we define the decoder by

$$\begin{aligned} \underline{\tilde{\mathbf{C}}}_{N-2} =& D_{N-1}(\underline{\mathbf{C}}_{N-1}), \\ &\vdots \\ \underline{\tilde{\mathbf{C}}}_1 = D_2(\underline{\tilde{\mathbf{C}}}_2), \quad \underline{\tilde{\mathbf{C}}}_0 =& D_1(\underline{\tilde{\mathbf{C}}}_1), \\ G(\mathbf{y}) =& D_0(\underline{\tilde{\mathbf{C}}}_0), \end{aligned} \tag{3.7}$$

where $\underline{\hat{\mathbf{C}}}_n$ is the n-th decoded signal, which is produced by the n-th decoder layer, yielding the general expression:

$$\underline{\tilde{\mathbf{C}}}_{n-1} = D_n(\underline{\tilde{\mathbf{C}}}_n) = \tilde{A}_{(\underline{\tilde{b}})}(f_{(s\uparrow)}(\underline{\tilde{\mathbf{C}}}_n) \circledast \underline{\tilde{\mathbf{K}}}_n^{\mathsf{T}}). \tag{3.8}$$

In the above, $\tilde{A}(\cdot)_{(\cdot)}$ is the activation function used in the decoder and $f_{(s\uparrow)}(\cdot)$ is an up-sampling function with factor s.

An important remark is that the encoder-decoder CNN does not always contain down/up-sampling layers in which case, the decimation factor s is unity, which causes $f_{(1\uparrow)}(\mathbf{x}) = f_{(1\downarrow)}(\mathbf{x}) = \mathbf{x}$ for any matrix $\mathbf{x}$. Furthermore, it should be noted also that we assume that the number of channels of the code $\underline{\mathbf{C}}_N$ is always larger than the previous one $\underline{\mathbf{C}}_{N-1}$. Furthermore, it should be noted that a single encoder layer $E_n(\cdot)$ and its corresponding decoder layer $D_n(\cdot)$ can be considered a single-layer encoder-decoder network/pair.

For completeness, we briefly discuss the dimensionality of the tensors of the encoding and decoding convolution layers (i.e. the number of channels). Specifically, the encoding convolution filter $\underline{\mathbf{K}}$ for a given layer has dimensions $(N_\mathrm{i} \times N_\mathrm{o} \times N_\mathrm{h} \times N_\mathrm{v})$, where N_i and N_o are the number of input and output channels for a convolution layer, respectively. Similarly, N_h and N_v are the number of elements in the horizontal and vertical directions, respectively. Note that the encoder increases the number of channels of the signal (e.g. $N_\mathrm{o} > N_\mathrm{i}$), akin to Ranzatto's design [92]. Furthermore, it is assumed that the *decoder* is symmetric in the number of channels to the *encoder*, so that the dimensions of the decoding convolution kernel $\tilde{\underline{\mathbf{K}}}^\mathsf{T}$ are $(N_\mathrm{o} \times N_\mathrm{i} \times N_\mathrm{h} \times N_\mathrm{v})$. The motivation of this symmetry is to emphasize the similarity between the signal processing and the CNN elements.

3.3 Fundamental concepts of signal processing

As shown by Ye *et al.* [40], within encoding-decoding CNNs, the signal is treated akin to well-known sparse representations, where the coefficients used for the transformation are directly learned from the training data. Prior to addressing this important concept in more detail, relevant supporting concepts such as sparsity, sparse transformations are explained, as well as non-linear signal estimation in the wavelet/framelet domain. Specifically, Section 3.3.1 defines the concept of sparsity, Section 3.3.2 addresses the concept of sparse transforms, such as the wavelet transform and singular value decomposition. Finally, Section 3.3.3 addresses the use of non-linear estimators on the framelet domain.

3.3.1 Sparsity

For a sparse signal, most coefficients are small and the relatively few large coefficients capture most of the information [106]. This characteristic allows to discard low-amplitude components with relatively small perceptual changes. Hereby, the use of sparse signals is attractive for applications such as image compression, denoising and suppression of artifacts.

Despite the convenient characteristics of sparse signals, natural images are often non-sparse. Still, there are numerous transformations, which allow to map the signal to a sparse domain and which are analogous to the internal operations of CNNs. For example, *singular value decomposition* factorizes the image in terms of two sets of orthogonal bases of which few basis pairs contain most of the energy of the image. An alternative transformation is based on *framelets*, where an image is decomposed in a multi-channel representation, whereby each resulting channel contains a fragment of the Fourier spectrum. In the remainder of this section, we will address these representations in more detail.

3.3.2 Sparse signal representations

A Singular value decomposition (SVD) and low-rank approximation

Assuming that an image (patch) is represented by a matrix $\mathbf{y}$ with dimensions $(N_\mathrm{r} \times N_\mathrm{c})$, where N_r and N_c are the number of rows and columns, respectively. Then, the singular value decomposition factorizes $\mathbf{y}$ as

$$\mathbf{y} = \sum_{n=0}^{N_\mathrm{SV}-1} (\underline{u}_n \underline{v}_n^\intercal) \cdot \underline{\sigma}[n], \tag{3.9}$$

in which N_SV is the number of singular values, n is a scalar index, while $\underline{u}_n$ and $\underline{v}_n$ are the n^th left- and right-singular vectors, respectively, having dimensions $(N_r \times 1)$ and $(1 \times N_c)$. Furthermore, vector $\underline{\sigma}$ contains the singular values and each of its entries $\underline{\sigma}[n]$ is the weight assigned to every basis pair $\underline{u}_n$, $\underline{v}_n$. This means that the product $(\underline{u}_n \underline{v}_n^\intercal)$ contributes more to the image content for higher values of $\underline{\sigma}[n]$. It is customary that the singular values are ranked in descending order and the amplitudes of the singular values $\underline{\sigma}$ are sparse, therefore $\underline{\sigma}[0] \gg \underline{\sigma}[N_\mathrm{SV} - 1]$. The reason for the sparsity of signals in the SVD-domain is because images intrinsically have a high correlation. For example, many images contain repetitive patterns (e.g. a wall with bricks, a fence, the tiles of a rooftop or the stripes of a zebra) or uniform regions (for example, the sky, the skin of a person). This means an image patch may contain only a few linearly independent vectors that describe most of the image contents. Consequently, a higher weight is assigned to those singular vectors/image bases.

Given that the amplitudes of the singular values of $\mathbf{y}$ in SVD are sparse, it is possible approximate $\hat{\mathbf{y}}$ with only a few vector products $\underline{u}_n \underline{v}_n^\intercal$. Note that this procedure reduces the rank of signal $\mathbf{y}$ and hence it is known as *low-rank approximation*. This process is equivalent to

$$\hat{\mathbf{y}} = \sum_{n=0}^{N_\mathrm{LR}-1} (\underline{u}_n \underline{v}_n^\intercal) \cdot \sigma[n], \tag{3.10}$$

where $N_\mathrm{LR} < N_\mathrm{SV}$. Note that this effectively cancels the terms $(\underline{u}_n \underline{v}_n^\intercal)$ for which the corresponding weight given by $\sigma[n]$ is low. Alternatively, it is possible to assign a weight of zero to the product $(\underline{u}_n \underline{v}_n^\intercal)$ for $n \geq N_\mathrm{LR}$ in Eq. (3.9).

The low-rank representation of a matrix is desirable for diverse applications among which we can find image denoising. The motivation for using low-rank approximation for this application results from the fact that –as mentioned earlier– natural images are considered low-rank due to the strong spatial correlation between pixels, whereas noise is high-rank (it is spatially uncorrelated). In consequence, reducing the rank/number of singular values decreases the presence of noise, while still providing a good approximation of the noise-free signal, as exemplified in Fig. 3.2.

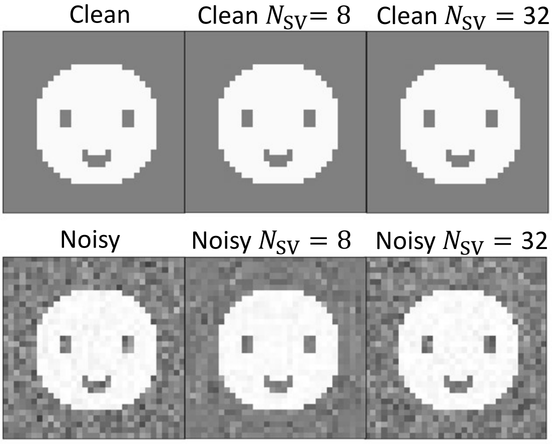

Clean
Clean N_{SV}= 8
Clean N_{SV} = 32
Noisy
Noisy N_{SV} = 8
Noisy N_{SV} = 32

Figure 3.2. SVD reconstruction of clean and corrupted images with a different number of singular values. Note that the reconstruction of the clean image with 8 or 32 singular values ($N_{\mathrm{SV}} = 8$ or $N_{\mathrm{SV}} = 32$, respectively) yields to reconstructions indistinguishable from the original image. This contrasts with their noisy counterparts, where $N_{\mathrm{SV}} = 8$ reconstructs a smoother image in which the noise is attenuated, while $N_{\mathrm{SV}} = 32$ reconstructs the noise texture perfectly.

B Framelets

Just as SVD, framelets are also commonly used for image processing. In a nutshell, a framelet transform is a signal representation that factorizes/decomposes an arbitrary signal into multiple bands/channels. Each of these channels contain a segment of the energy of the original signal. In image and signal processing, the framelet bands are the result of convolving the analyzed signal with a group of discrete filters that have finite length/support. In this text, the most important characteristic that the filters of the framelet transform should comply with, is that the bands they generate capture *all* the energy contained on the input to the decomposition. This is important to avoid the loss of information of the decomposed signal. In this text, we refer to framelets that comply with the previous characteristics as *tight* framelets and the following paragraphs will describe this property in more detail.

In its decimated version, the framelet decomposition for *tight* frames is represented by

$$\underline{\mathbf{Y}}_{\mathrm{fram}} = f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{F}}), \tag{3.11}$$

in which $\underline{\mathbf{Y}}_{\mathrm{fram}}$ is the decomposed signal and $\underline{\mathbf{F}}$ is the framelet basis (tensor). Note that the signal $\underline{\mathbf{Y}}_{\mathrm{fram}}$ has more channels than $\mathbf{y}$. Furthermore, the original signal $\mathbf{y}$ is recovered from $\mathbf{Y}_{\mathrm{fram}}$ by

$$\mathbf{y} = f_{(2\uparrow)}(\mathbf{Y}_{\mathrm{fram}}) \circledast \underline{\tilde{\mathbf{F}}}^{\mathsf{T}} \cdot c. \tag{3.12}$$

Here, $\underline{\tilde{\mathbf{F}}}$ is the filter of the inverse framelet transform and c denotes an arbitrary constant. If $c = 1$ the framelet is *normalized*. Finally, note that the framelet transform can also be *undecimated*. This means that in undecimated representations, the down-sampling and up-sampling layers $f_{(2\downarrow)}(\cdot)$ and $f_{(2\uparrow)}(\cdot)$ are not used. An important

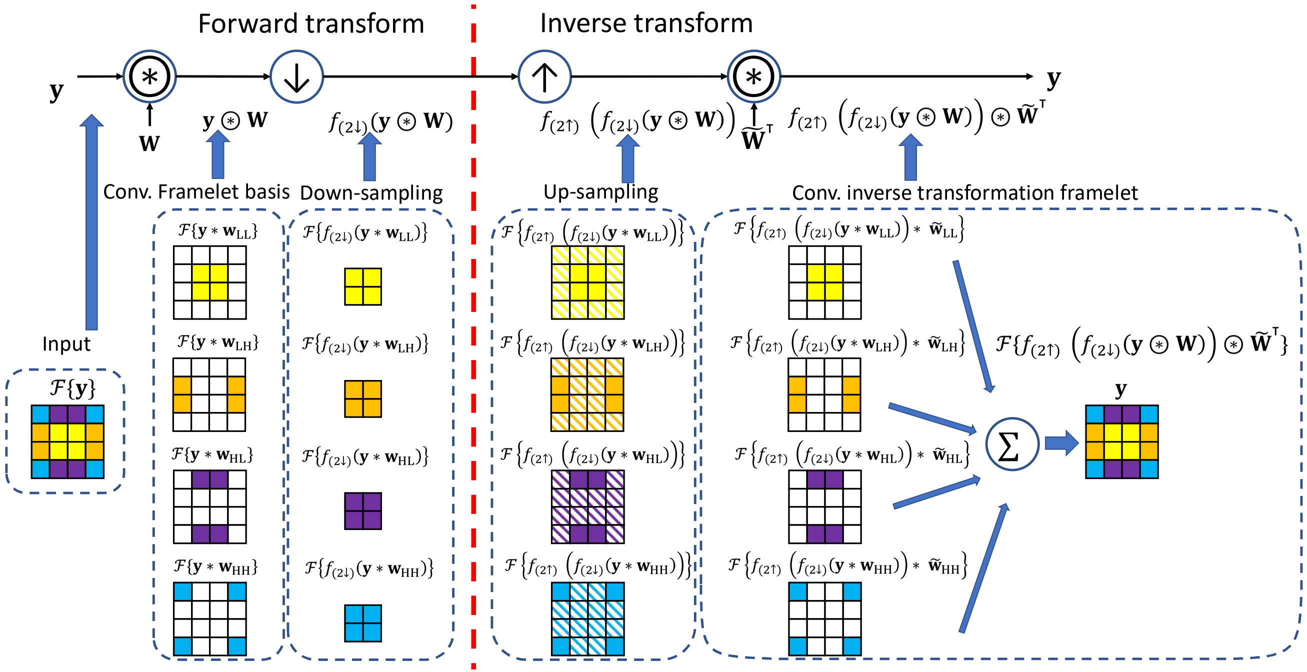

Forward transform
Inverse transform
y
W
y ⊛ W
f(2↓)(y ⊛ W)
f(2↑)(f(2↓)(y ⊛ W))
W̃ᵀ
f(2↑)(f(2↓)(y ⊛ W)) ⊛ W̃ᵀ
y
Input
ℱ{y}
Conv. Framelet basis
Down-sampling
Up-sampling
Conv. inverse transformation framelet
ℱ{y * w_LL}
ℱ{y * w_LH}
ℱ{y * w_HL}
ℱ{y * w_HH}
ℱ{f(2↓)(y * w_LL)}
ℱ{f(2↓)(y * w_LH)}
ℱ{f(2↓)(y * w_HL)}
ℱ{f(2↓)(y * w_HH)}
ℱ{f(2↑)(f(2↓)(y * w_LL))}
ℱ{f(2↑)(f(2↓)(y * w_LH))}
ℱ{f(2↑)(f(2↓)(y * w_HL))}
ℱ{f(2↑)(f(2↓)(y * w_HH))}
ℱ{f(2↑)(f(2↓)(y * w_LL)) * w̃_LL}
ℱ{f(2↑)(f(2↓)(y * w_LH)) * w̃_LH}
ℱ{f(2↑)(f(2↓)(y * w_HL)) * w̃_HL}
ℱ{f(2↑)(f(2↓)(y * w_HH)) * w̃_HH}
Σ
ℱ{f(2↑)(f(2↓)(y ⊛ W)) ⊛ W̃ᵀ}
y

Figure 3.3. 2D spectrum analysis of the decimated discrete framelet decomposition and reconstruction. In the figure, function $\mathcal{F}\{\cdot\}$ stands for the amplitude Fourier spectrum of the input argument. The yellow squares indicate a region in the low-frequency area of the Fourier spectrum, while the purple, orange and blue squares indicate the high-pass/detail bands. For these images, ideal orthogonal bases are assumed. Note that the forward transform is composed by two steps. First, the signal is convolved with the wavelet basis $(\mathbf{y} \circledast \underline{\mathbf{W}})$. Afterwards, down-sampling is applied to the signal $(f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{W}}))$. During the inverse transformation, the signal is up-sampled by inserting zeros between each sample $(f_{(2\uparrow)}(f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{W}})))$, which causes spatial aliasing (dotted blocks). Finally, the spatial aliasing is removed by the inverse transform filter $\underline{\tilde{\mathbf{W}}}$ and all the channels are added $(f_{(2\uparrow)}(f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{W}})) \circledast \underline{\mathbf{W}}^{\intercal})$.

property of the undecimated representation is that it is less prone to aliasing than its decimated counterpart, but more computationally expensive. Therefore, for efficiency reasons, the decimated framelet decomposition is often preferred over the undecimated representation. In summary, the decomposition and synthesis of the decimated framelet decomposition is represented by

$$\mathbf{y} = f_{(2\uparrow)}\big(f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{F}})\big) \circledast \underline{\tilde{\mathbf{F}}}^{\mathsf{T}} \cdot c, \tag{3.13}$$

while for the undecimated framelet it holds that

$$\mathbf{y} = (\mathbf{y} \circledast \underline{\mathbf{F}}) \circledast \underline{\tilde{\mathbf{F}}}^{\mathsf{T}} \cdot c. \tag{3.14}$$

A notable normalized framelet is the *discrete wavelet transform* (DWT), where variables $\underline{\mathbf{F}}$ and $\underline{\tilde{\mathbf{F}}}$ are replaced by tensors $\underline{\mathbf{W}} = \big(\mathbf{w}_{\mathrm{LL}}, \mathbf{w}_{\mathrm{LH}}, \mathbf{w}_{\mathrm{HL}}, \mathbf{w}_{\mathrm{HH}}\big)$ and $\underline{\tilde{\mathbf{W}}} = \big(\tilde{\mathbf{w}}_{\mathrm{LL}}, \tilde{\mathbf{w}}_{\mathrm{LH}}, \tilde{\mathbf{w}}_{\mathrm{HL}}, \tilde{\mathbf{w}}_{\mathrm{HH}}\big)$, respectively. Here, $\mathbf{w}_{\mathrm{LL}}$ is the filter for the low-frequency band, while $\mathbf{w}_{\mathrm{LH}}$, $\mathbf{w}_{\mathrm{HL}}$, $\mathbf{w}_{\mathrm{HH}}$ are the filters used to extract the detail in the horizontal, vertical and diagonal directions, respectively. Finally, $\tilde{\mathbf{w}}_{\mathrm{LL}}$ $\tilde{\mathbf{w}}_{\mathrm{LH}}$, $\tilde{\mathbf{w}}_{\mathrm{HL}}$, $\tilde{\mathbf{w}}_{\mathrm{HH}}$ are the filters of the inverse decimated DWT.

In order to understand the DWT more intuitively, Fig. 3.3 shows the decimated framelet decomposition using the filters of the discrete wavelet transform. Note that the convolution $\mathbf{y} \circledast \underline{\mathbf{W}}$ results in a four-channel signal, where each channel contains only a fraction of the spectrum of image $\underline{\mathbf{y}}$. This allows to down-sample each channel with minimal aliasing. Furthermore, to recover the original signal, each individual channel is up-sampled, thereby introducing aliasing, which is then removed by the filters of the inverse transform. Finally, all the channels are added and the original signal is recovered.

Analogous to the low-rank approximation, in framelets, the reduction of noise is achieved by setting noisy components to zero. These components are typically assumed to have low-amplitude compared with the amplitude of the sparse signal, as expressed by

$$\hat{\mathbf{y}} = f_{(2\uparrow)}\big(\tau_{(\underline{t})}\big(f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{F}})\big)\big) \circledast \underline{\tilde{\mathbf{F}}}^{\mathsf{T}} \cdot c, \tag{3.15}$$

where $\tau_{\underline{t}}(\cdot)$ is a generic thresholding/shrinkage function, which sets each of the pixels in $f_{(2\downarrow)}(\mathbf{y} \circledast \underline{\mathbf{F}})$ to zero when values are lower than the threshold level $\underline{t}$.

3.3.3 Nonlinear signal estimation in the framelet domain

As mentioned in Section 3.3.2, framelets decompose a given image $\mathbf{y}$ by convolving it with a tensor $\underline{\mathbf{F}}$. Note that many of the filters that compose $\underline{\mathbf{F}}$ have a high-pass nature. Images often contain approximately uniform regions in which the variation is low, therefore, convolving a signal $\mathbf{y}$ with a high-pass filter $\mathbf{f}_{\mathrm{h}}$ –where $\mathbf{f}_{\mathrm{h}} \in \underline{\mathbf{F}}$– produces the sparse detail band $\mathbf{d} = \mathbf{f}_{\mathrm{h}} * \mathbf{y}$ in which uniform regions have low amplitudes, while transitions i.e. edges contain most of the energy of the bands.

Assuming a model in which a single pixel $d \in \mathbf{d}$ is observed, which is contaminated with additive noise η. Then, the resulting observed pixel z is defined by

$$z = d + \eta. \tag{3.16}$$

In order to recover the noiseless pixel d from observation z, it is possible to use the point-*maximum a posteriori* (MAP) estimate [14], [76], defined by the maximization problem

$$\hat{d} = \arg\max_{d} \big[\ln \big(P(d|z) \big) \big]. \tag{3.17}$$

Here, the log-posterior $\ln \big(P(d|z) \big)$ is defined by

$$\ln \big(P(d|z) \big) = \ln \big(P(z|d) \big) + \ln \big(P(d) \big), \tag{3.18}$$

where the conditional probability density function (PDF) $P(z|d)$ expresses the noise distribution, which is often assumed Gaussian and is defined by

$$P(z|d) \propto \exp \left(- \frac{(z-d)^2}{2\sigma_\eta^2} \right). \tag{3.19}$$

Here, σ_η^2 is the noise variance. Furthermore, as prior probability, it is assumed that the distribution of $P(d)$ corresponds to a Laplacian distribution, which has been used in wavelet-based denoising [14]. Therefore, $P(d)$ is mathematically described by

$$P(d) \propto \exp \left(- \frac{|d|}{\sigma_d} \right), \tag{3.20}$$

where σ_d is the dispersion measure of the Laplace distribution. For reference, Fig. 3.4 portrays an example of a Gaussian and a Laplacian PDF. Note that the Laplacian distribution has a higher probability of zero elements to occur than the Gaussian distribution for the same standard deviation. Finally, substituting Eq. (3.19) and Eq. (3.20) in Eq (3.18) results in

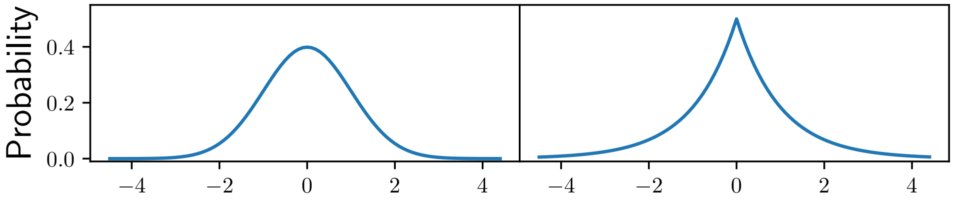


Figure 3.4. Probability density function for Gaussian (left) and Laplacian (right) distributions.

$$\ln \big(P(d|z) \big) \propto - \frac{(z-d)^2}{2\sigma_\eta^2} - \frac{|d|}{\sigma_d}. \tag{3.21}$$

In the above, maximizing d in $\ln \big(P(d|z) \big)$ with the first derivative criterion –in an (un) constrained way– leads to two common activations in noise reduction CNNs: the ReLU and the soft-shrinkage function. Furthermore, the solution also can be used to derive the so-called *clipping* function, which is useful in residual networks.

For reference and further understanding, Fig. (3.5) portrays the elements composing the noise model of Eq. (3.16), the signal transfer characteristics of the ReLU, soft-shrinkage and clipping functions, and the effect that these functions have on the signal of the observed noisy detail band z.

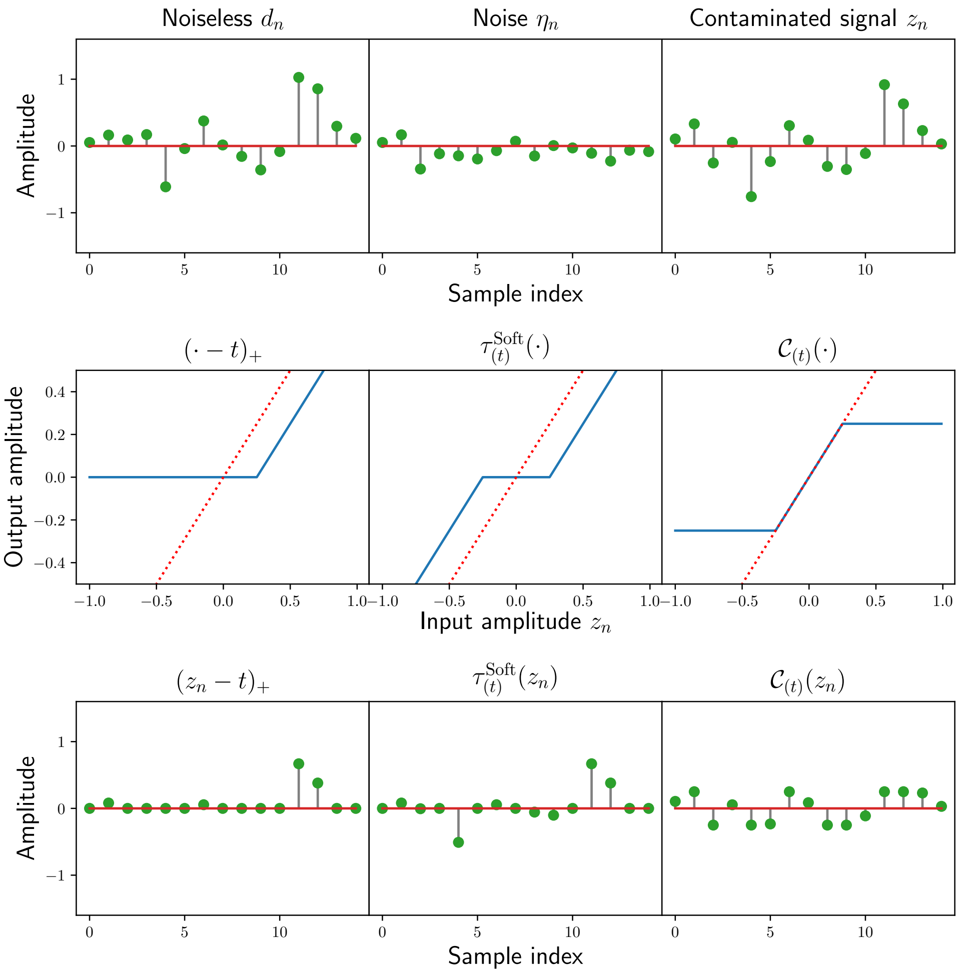


Figure 3.5. Signals involved in the additive noise model, input/output transfer characteristics of activation layers and estimates produced by the activation layers when applied to the noise-contaminated signal. The first row shows the signals involved in the additive noise model. The second row depicts the output amplitude of activation functions with respect to the input amplitude. Finally, the last row depicts the application of the activation functions to the noisy observation z.

A Rectified linear unit (ReLU)

If Eq. (3.21) is solved for d while constraining the estimator to be positive, the noiseless estimate $\hat{d}$ becomes

$$\hat{d} = (z - t)_+, \tag{3.22}$$

which is also expressed by

$$(z - t)_+ = \begin{cases} z - t, & \text{if } z \geq t, \\ 0, & \text{if } t > z. \end{cases} \tag{3.23}$$

Here, the threshold level is defined by

$$t = \sigma_\eta^2 / \sigma_d. \tag{3.24}$$

Note that this estimator cancels the negative and low-amplitude elements of d lower than the magnitude of the threshold level t. For example, if the signal content on the feature map is low, then $\sigma_d \to 0$. In such case, $t \to +\infty$ and consequently $\hat{d} \to 0$. This means that the channel is suppressed. Alternatively, if the feature map has strong signal presence i.e. $\sigma_d \to \infty$, consequently $t \to 0$ and then $\hat{d} \to (z)_+$.

A final remark is made on the modeling of functions of a CNN. It should be noted that the estimator of Eq. (3.22) is analogous to the activation function of a CNN, known as *rectified linear unit* (ReLU). However, in a CNN the value of t would be the bias b learned from the training data.

B Soft-shrinkage/thresholding

If Eq. (3.21) is maximized in an unconstrained way, the estimate $\hat{d}$ is

$$\hat{d} = \tau_{(t)}^{\text{Soft}}(z) = (z - t)_+ - (-z - t)_+. \tag{3.25}$$

Here, $\tau_{(t)}^{\text{Soft}}(\cdot)$ denotes the soft-shrinkage/-thresholding function, which is often also written in the form

$$\tau_{(t)}^{\text{Soft}}(z) = \begin{cases} z + t, & \text{if } z \geq t, \\ 0, & \text{if } t > z \geq -t, \\ z - t, & \text{if } -t > z. \end{cases} \tag{3.26}$$

It can be observed that the soft-threshold enforces the low-amplitude components whose magnitude is lower than the magnitude threshold level t to zero. In this case, t is also defined by Eq. (3.24). It should be noted that the soft-shrinkage estimator can also be obtained from a variational perspective [107]. Finally, it can be observed that the soft-shrinkage is the superposition of two ReLU functions, which has been pointed out by Fan *et al.* [39].

C Soft-clipping

In Section 3.3.3-A and Section 3.3.3-B, the estimate $\hat{d}$ is obtained directly from the noisy observation z. Alternatively, it is possible to estimate the noise η and subtract it from z akin to the residual CNNs represented by Eq. (3.3). This can be achieved by solving the model

$$\hat{\eta} = z - \hat{d} = z - \tau_{(t)}^{\text{Soft}}(z), \tag{3.27}$$

which is equivalent to

$$\hat{\eta} = \mathcal{C}_{(t)}^{\text{Soft}}(z) = z - ((z-t)_+ - (-z-t)_+), \tag{3.28}$$

where $\mathcal{C}_{(t)}^{\text{Soft}}(\cdot)$ is the *soft-clipping* function. Note that this function also can be expressed by

$$\mathcal{C}_{(t)}^{\text{Soft}}(z) = \begin{cases} t, & \text{if } z \geq t, \\ z, & \text{if } t \geq z > -t, \\ -t, & \text{if } -t \geq z. \end{cases} \tag{3.29}$$

D Other thresholding layers

One of the main drawbacks of the soft-threshold activation is that it is a biased estimator. This limitation has been addressed by the hard [60] and semi-hard [108] thresholds, which are (asymptotically) unbiased estimators for large input values. In this section, we focus solely on the semi-hard threshold and avoid the hard variant, because it is discontinuous and, therefore, not suited for models that rely on gradient-based optimization, such as CNNs.

Among the semi-hard thresholds, two notable examples are the *garrote shrink* and the shrinkage functions generated by derivatives of Gaussians (DoG) [58], [108]. The garrote shrink function $\tau_{(\cdot)}^{\text{Gar}}(\cdot)$ is defined by

$$\tau_{(t)}^{\text{Gar}}(z) = \frac{(z^2 - t^2)_+}{z}. \tag{3.30}$$

Furthermore, an example of a shrinkage function based on the derivative of Gaussians is given by

$$\tau_{(t)}^{\text{DoG}}(z) = z - \mathcal{C}_{(t)}^{\text{DoG}}(z), \tag{3.31}$$

where the semi-hard clipping function with the derivative of Gaussians $\mathcal{C}_{(\cdot)}^{\text{DoG}}(\cdot)$ is given by

$$\mathcal{C}_{(t)}^{\text{DoG}}(z) = z \cdot \exp\left(-\frac{z^p}{t^p}\right), \tag{3.32}$$

in which p is an even number.

The garrote and semi-hard DoG shrinkage functions are shown in Fig. 3.6, as well as their clipping counterparts. Note that the shrinkage functions approximate unity for $|z| \gg t$. Therefore, they are asymptotically unbiased for large signal values.

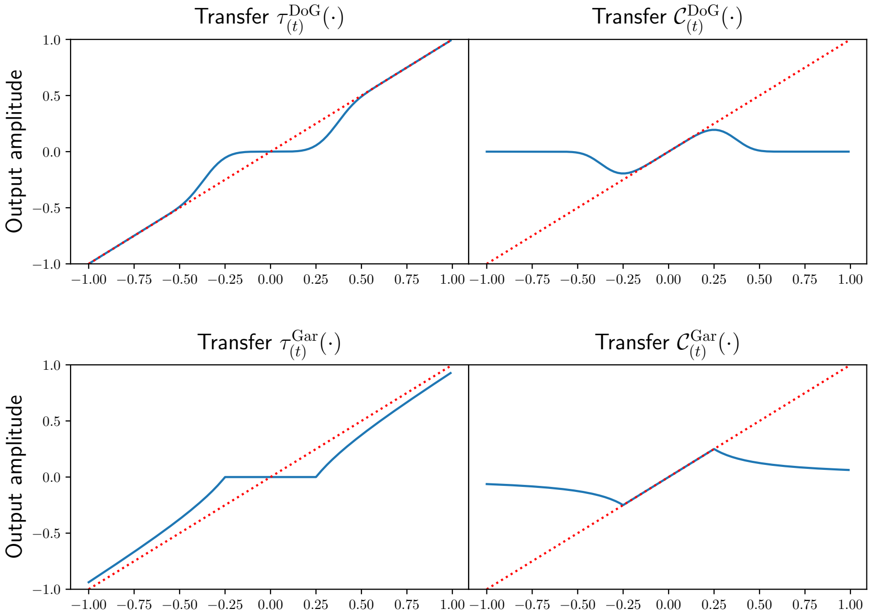


Figure 3.6. Transfer characteristics of the semi-hard thresholds based on the derivative of Gaussians (first row) and of the garrote shrink (second row), as well as their clipping counterparts. Note that in contrast with the soft-shrinkage and soft-clipping functions shown in Fig. 3.5, the semi-hard thresholds tend to unity for large values, while the semi-hard clipping functions tend to zero for large signal intensities.

The final thresholding function addressed in this section is the linear expansion of thresholds, as proposed by Blu and Luisier [108]. This technique combines multiple thresholding functions to improve the performance. This approach is known as *linear expansion of thresholds* (LET) and it is defined by

$$\tau^{\text{LET}}_{(\underline{t})}(z) = \sum_{n=0}^{N_T-1} a_n \cdot \tau_{(t_n)}(z), \tag{3.33}$$

where a_n is the weighting factor assigned to each threshold, where all weighting factors add up to unity.

3.4 Bridging the gap between signal processing and CNNs: Deep convolutional framelets and shrinkage-based CNNs

This section addresses the theoretical operation of noise reduction convolutional neural networks, based on ReLUs and shrinkage/thresholding functions. The first part of this section describes the theory of deep convolutional framelets [40], which

–up to this moment– is the most extensive study on the operation of encoding-decoding ReLU-based CNNs. Afterwards, the section concentrates on the operation of networks which use shrinkage functions instead of ReLUs [39], [48], [58], with the aim of mimicking well-established denoising algorithms [14], [60] by assuming that the encoder learns to perform a sparse transformation where soft shrinkage is applied akin to the soft-shrinkage denoising algorithm first introduced in Section 2.1.3. Finally, the last part of this section addresses the connections between both methods and additional links between convolutional neural networks and signal processing.

3.4.1 Theory of deep convolutional framelets

A Description of this theory

The theory of deep convolutional framelets [40] describes the operation of encoding-decoding ReLU-based CNNs. Its most relevant contributions are as follows. (1) It establishes the equivalence of framelets and the convolutional layers of CNNs. (2) The theory of deep convolutional framelets provides the conditions to preserve the signal integrity within a ReLU CNN. (3) It explains how ReLU and convolution layers reduce noise within an encoding-decoding CNN. This powerful theory is inspired by the low-rank denoising methods, based on the Hankel-matrix decomposition such as ALOHA presented in Section 2.1.4.

The similarity between framelets and the encoding and decoding convolutional filters can be observed when comparing Eqs. (3.5), (3.8) with Eqs. (3.11), (3.12), where it becomes visible that the convolution structure of encoding-decoding CNNs is analogous to the forward and inverse framelet decomposition.

Regarding the signal reconstruction characteristics, the theory of deep convolutional framelets states the following. In order to be able to recover an arbitrary signal $\mathbf{y} \in \mathbb{R}^N$, the number of output channels of a convolution layer with ReLU activation should *at least* duplicate the number of input channels. Furthermore, the convolution kernel should be composed of filter pairs with opposite phase. These two requirements ensure that any negative and positive values propagate through the network. Under these conditions, the encoding and decoding convolution filters $\underline{\mathbf{K}}$ and $\underline{\tilde{\mathbf{K}}}$ should comply with

$$\mathbf{y} = (\mathbf{y} \circledast \underline{\mathbf{K}})_+ \circledast \underline{\tilde{\mathbf{K}}} \cdot c. \tag{3.34}$$

It can be noticed that Eq. (3.34) is an extension of Eq. (3.14), which describes the reconstruction characteristics of tight framelets. From this point, we refer to convolutional kernels compliant with Eq (3.34) as *phase-complementary tight framelets*. As a final remark, it should be noted that a common practice in CNN designs is also to use ReLU non-linearities in the decoder, in such case the phase-complementary tight-framelet condition can still be met as long as the pixels $y \in \mathbf{y}$ comply with $y \geq 0$, which is equivalent to

$$\mathbf{y} = (\mathbf{y})_+ = \Big((\mathbf{y} \circledast \underline{\mathbf{K}})_+ \circledast \underline{\tilde{\mathbf{K}}} \cdot c\Big)_+. \tag{3.35}$$

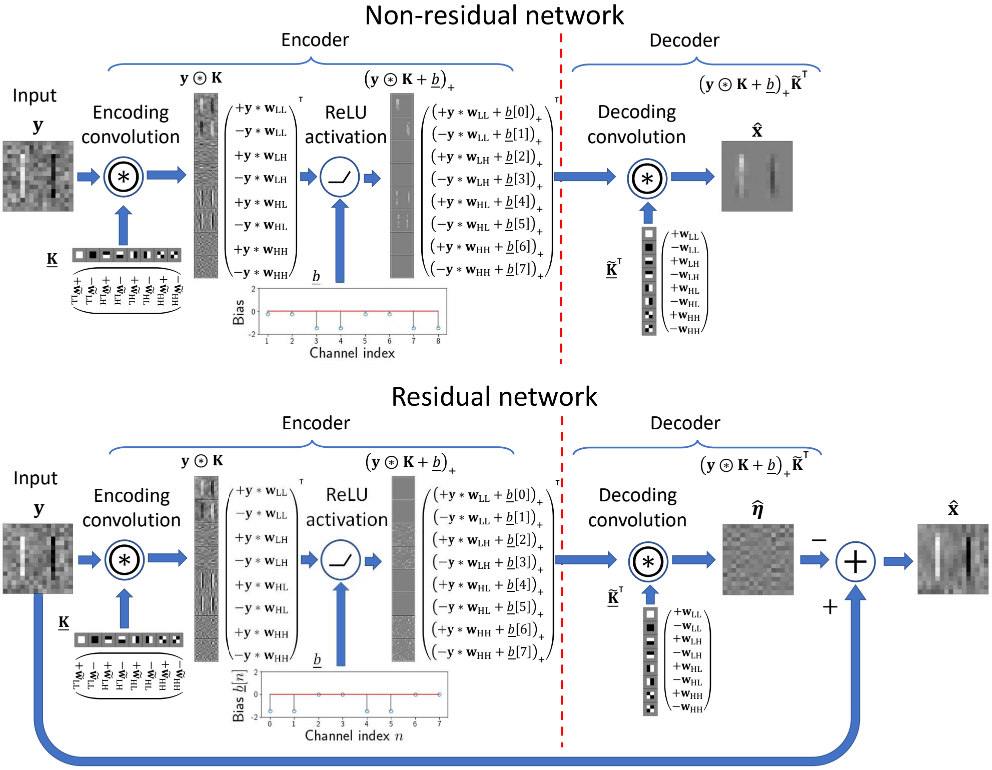


Figure 3.7. Operation of a denoising (non-) residual ReLU CNN according to the theory of deep convolutional framelets (TDCF). In the figure, the noisy observation $\mathbf{y}$ is composed by two vertical bars plus uncorrelated Gaussian noise. Furthermore, for this example, the encoding and decoding convolution filters ($\underline{\mathbf{K}}$ and $\underline{\tilde{\mathbf{K}}}$, respectively) are the Haar basis of the 2D discrete wavelet transform and its phase-inverted counterparts. Given the content of the image, the image in the decomposed domain $\mathbf{y} \circledast \underline{\mathbf{K}}$ produces only a weak activation for the vertical and diagonal filters ($\mathbf{w}_{\mathrm{LH}}$ and $\mathbf{w}_{\mathrm{HH}}$, respectively) and those feature maps contain mainly noise. In the case of the non-residual network, the ReLUs and biases suppress the channels with low activation (see column $(\mathbf{y} \circledast \underline{\mathbf{K}} + \underline{b})_+$), which is akin to the low-rank approximation. In contrast, in the residual example, the channels with image content are suppressed, while preserving the uncorrelated noise. Finally, the decoding section reconstructs the noise-free estimate $\tilde{\mathbf{x}}$ for the non-residual network or the noise estimate $\hat{\boldsymbol{\eta}}$ for the residual example, where it is subtracted from $\mathbf{y}$ to compute the noiseless estimate $\hat{\mathbf{x}}$.

It can be observed that the relevance of the properties defined in Eqs. (3.34) and (3.35) is that they ensure that a CNN can propagate any arbitrary signal, which is important to avoid any distortions (such as image blur) in the processed images.

An additional element of the theory of deep convolutional framelets regarding the reconstruction of the signal, is to show that conventional pooling layers (e.g. average pooling) discard high-frequency information of the signal, which effectively blurs the processed signals. Furthermore, Ye *et al.* [40] have demonstrated that this can be fixed by replacing the conventional up/down-sampling layers by reversible operations, such as the discrete wavelet transform. To exemplify this property, we refer to Fig. 3.3. If only an average pooling layer followed by an up-sampling stage would be applied, the treatment of the signal would be equivalent to the low-frequency branch of the DWT. Consequently, only the low-frequency spectrum of the signal would be recovered and images processed with that structure would become blurred. In contrast, if the full forward and inverse wavelet transform of Fig. 3.3 is used for up/down-sampling, it is possible to reconstruct any signal, irrespective of its frequency content.

The ultimate key contribution of the theory of deep convolutional framelets is the explanation of the operation of ReLU-based noise reduction CNNs. For the non-residual configuration, ReLU CNNs perform the following sequence of operations. (1) The convolution filters decompose the incoming signal into a sparse multi-channel representation. (2) The feature maps which are uncorrelated to the signal, contain mainly noise. In this case, the bias and the ReLU activation cancel the noisy feature maps in a process analogous to the MAP estimate shown in Section 3.3.3-A. (3) The decoder reconstructs the filtered image. In the case of *residual* networks, the CNN learns to estimate the noise, which means that in that configuration the ReLU non-linearities suppress the channels with high activation.

Fitting low-rank approximation in ReLU CNNs

From low-rank approximation to an encoding-decoding CNN. In order to further understand the analogy between CNNs and low-rank approximation established by the theory of deep convolutional framelets, we can use as starting point the definition of singular value decomposition, which is expressed in Eq. (3.9), by

$$\mathbf{y} = \sum_{n=0}^{N_{\mathrm{SV}}-1} (\underline{u}_n \underline{v}_n^{\intercal}) \cdot \underline{\sigma}[n].$$

Given that left and right singular vector pairs $\underline{u}_n \underline{v}_n^{\intercal}$ generate an image $\mathbf{D}[n]$, then Eq. (3.9) can be rewritten to

$$\mathbf{y} = \sum_{n=0}^{N_{\mathrm{SV}}-1} \underline{\mathbf{D}}[n] \cdot \underline{\sigma}[n], \tag{3.36}$$

where tensor $\underline{\mathbf{D}} = \big((\underline{u}_0 \underline{v}_0^{\intercal}) \ldots (\underline{u}_{N_{\mathrm{SV}}-1} \underline{v}_{N_{\mathrm{SV}}-1}^{\intercal})\big)$ contains the products of the left and right singular vectors and has dimensions $(1 \times N_{\mathrm{SV}} \times M \times N)$. Furthermore, the preceding equation can be further reformulated to

$$\mathbf{y} = \underline{\mathbf{D}} \circledast \underline{\tilde{\mathbf{K}}}^{\intercal}, \tag{3.37}$$

in which $\underline{\tilde{\mathbf{K}}}^{\intercal} = \big((\underline{\sigma}[0]) \ldots (\underline{\sigma}[N_{\mathrm{SV}-1}])\big)^{\intercal}$, where the brackets of the (1×1) filters have been excluded for simplicity. In addition, it is can be assumed that for performing low-rank approximation of signal $\mathbf{y}$ it is attractive to use the reformulation of Eq. (3.37). If we assume that $\underline{\mathbf{D}} \in \mathbb{R}^{N}_{\geq 0}$, then the low-rank approximation of $\mathbf{y}$ into $\hat{\mathbf{y}}$ can be expressed by

$$\hat{\mathbf{y}} = (\mathbf{D} + \underline{b})_+ \circledast \underline{\tilde{\mathbf{K}}}^\intercal, \tag{3.38}$$

in which the values $\underline{b}$ are set to zero for the channels of $\underline{\mathbf{D}}$ that have high contributions to the image content. Conversely, the channels of $\underline{\mathbf{D}}[n]$ with less perceptual relevance are then cancelled by assigning large negative values to the corresponding entries of $\underline{b}$. As final reformulation, we can assume that the basis images $\underline{\mathbf{D}}$ are the result of decomposing our input image $\mathbf{y}$ with a set of feature maps generated by the encoder of a CNN i.e. $\underline{\mathbf{D}} = \mathbf{y} \circledast \underline{\mathbf{K}}$, which transforms Eq. (3.38) into

$$\hat{\mathbf{x}} = (\mathbf{y} \circledast \underline{\mathbf{K}} + \underline{b})_+ \circledast \underline{\tilde{\mathbf{K}}}^\intercal. \tag{3.39}$$

Here, it is visible that Eq. (3.39) is analogous to the encoding-decoding architecture defined in Eqs. (3.4) to (3.8) and the encoder and decoder filters are akin to the framelet formulation of Section 3.3.2-B. Note that Eq. (3.39) assumes that the entries $\underline{\mathbf{D}} = \mathbf{y} \circledast \underline{\mathbf{K}}$ are positive, which may be not always true. In this situation, tensor $\underline{\mathbf{D}}$ requires redundant channels in which their respective phases are inverted to avoid the signal loss. Furthermore, it should also be noticed that in a CNN, the bias/threshold level is not inferred from the statistics of the feature maps, but learned from the data presented to the network during training.

Multi-layer designs. It should be noted that CNNs contain multiple layers, which recursively decompose/reconstruct the signal. This may pose an advantage with respect to conventional low-rank approximation algorithms for a few reasons. First, the data-driven nature of CNNs allows to learn the basis functions which optimally decompose and suppress noise in the signal. Second, since networks are deep, the incoming signal is recursively decomposed and sparsified. This multi-decomposition scheme is very similar to the designs used in noise-reduction algorithms based on framelets. It can be noted that the recursive sparsifying principles have been observed in the past in methods such as the (learned) iterative soft-thresholding algorithm [109], [110] as well as convolutional sparse coding. In fact, convolutional sparse-coding approach, which also has been used for interpreting the operation of CNNs [41]

What about practical implementations? When training a CNN, the parameters of the model (i.e. $\mathbf{K}$, $\tilde{\mathbf{K}}^\intercal$ and $\underline{b}$) are updated to reduce the loss between the processed noisy signal and the ground truth, which does not warranty that the numerical values of the convolution filters and biases of the trained model comply with the assumptions performed here. This is because CNNs do not have mechanisms to enforce that filters have properties such as sparsity or perfect reconstruction and negative values for the biases. Consequently, CNNs may not necessarily perform a low-rank approximation of the signal, although the mathematical formulation of the low-rank approximation and the single-layer encoding-decoding are similar. Hence, the analysis presented here should be treated as insight on the mathematical formulation and/or potential properties that can be enforced for specific applications and not as a literal description of what trained models do.

3.4.2 Shrinkage and clipping-based CNNs

Just as ReLU networks, the encoder of shrinkage networks [39], [48], [58] separates the input signal in a multi-channel representation. As a second processing stage, the shrinkage networks estimate the noiseless encoded signal by cancelling the low-amplitude pixels in the feature maps in a process akin to the MAP estimate of Section 3.3.3-B. As final step, the encoder reconstructs the estimated noiseless image. Note that the use of shrinkage functions reduces the number of channels required by ReLU counterparts to achieve perfect signal reconstruction, because the shrinkage activation preserves positive and negative values, while ReLU only preserves the positive part of the signal.

As shown in Section 3.2.1, in residual learning, a given encoding-decoding network estimates the noise signal $\boldsymbol{\eta}$, so that it can be subtracted from the noisy observation $\mathbf{y}$ to generate the noiseless estimate $\hat{\mathbf{x}}$. Furthermore, as discussed in Section 3.3.3-C, in the framelet domain this is achieved by preserving the low-amplitude values of the feature maps by clipping the signal. Consequently, in residual networks, the shrinkage functions are explicitly replaced by clipping activations. This contrasts with the ReLU networks, where the same activation suffices for both, residual and non-residual designs.

Visual examples of the operation of single-layer shrinkage and clipping networks are presented in Fig. 3.7, where it can be noted that the operation of shrinkage and clipping networks is analogous to their ReLU counterparts, with the main difference that shrinkage and clipping networks do not require phase-complements in the encoding and decoding layers as ReLU-based CNNs do.

3.4.3 Shrinkage and clipping in ReLU networks

As addressed in Section 3.3.3, the soft-threshold function is the superposition of two ReLU activations. As a consequence, it is feasible that in ReLU CNNs shrinkage behavior can arise in addition to the low-rankness enforcement mentioned in Section 3.4.1. It should be noted that this can happen only if the number of channels of the encoder and decoder complies with the redundancy constraints of the theory of deep convolutional framelets and if the decoder is linear. To prove this, Eq. (3.25) is re-parameterized as

$$\hat{\mathbf{d}} = (\mathbf{y} \circledast \underline{\mathbf{K}} + \underline{b})_{+} \circledast \underline{\tilde{\mathbf{K}}}^{\intercal}, \tag{3.40}$$

where convolution filters $\underline{\mathbf{K}}$ and $\underline{\tilde{\mathbf{K}}}^{\intercal}$ are defined by

$$\underline{\mathbf{K}} = \underline{\tilde{\mathbf{K}}} = \begin{pmatrix} \begin{pmatrix} \mathbf{I} & -\mathbf{I} \end{pmatrix} \end{pmatrix} \tag{3.41}$$

and

$$\underline{b} = \begin{pmatrix} -t & -t \end{pmatrix}, \tag{3.42}$$

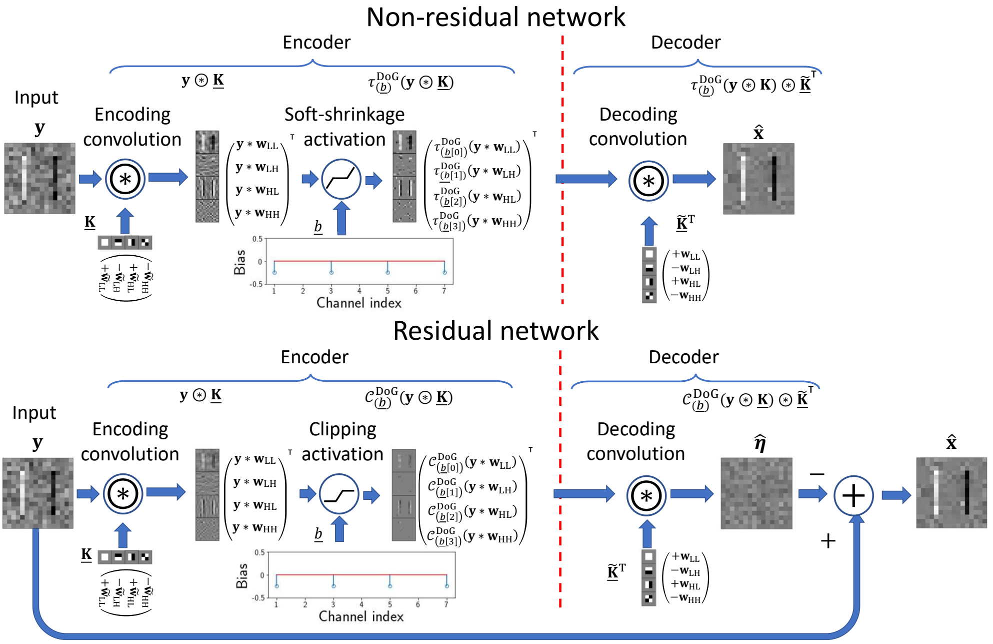


Figure 3.8. Operation of denoising in shrinkage and clipping networks. In the non-residual configuration, the noisy signal $\mathbf{y}$ is decomposed by a set of convolution filters, which for this example are the 2D Haar basis functions of the discrete wavelet transform ($\mathbf{y} \circledast \underline{\mathbf{K}}$). As a second step, the semi-hard shrinkage produces a MAP estimate of the noiseless detail bands/feature maps ($\tau^{\text{DoG}}_{(\underline{b})}(\mathbf{y} \circledast \underline{\mathbf{K}})$). As third and final step, the decoder maps the estimated noiseless encoded signal to the original image domain. In the residual network, the behavior is similar, but the activation layer is a clipping function that performs a MAP estimate of the noise in the feature maps, which is reconstructed by the decoder to generate the noise estimate $\hat{\boldsymbol{\eta}}$. After reconstruction, the noise estimate is subtracted from the noisy observation $\mathbf{y}$ to generate the noise-free estimate $\hat{\mathbf{x}}$.

where $\underline{b}$ represents the threshold value.

In addition to the soft-shrinkage, note that the clipping function described by Eq. (3.28) can be also expressed by Eq. (3.40), when $\underline{\mathbf{K}}$, $\underline{\tilde{\mathbf{K}}}^{\mathsf{T}}$ and $\underline{b}$ are defined by

$$\begin{aligned} \underline{\mathbf{K}} &= \left(\begin{pmatrix}\mathbf{I} & -\mathbf{I} & \mathbf{I} & -\mathbf{I}\end{pmatrix}\right), \\ \underline{\tilde{\mathbf{K}}}^{\mathsf{T}} &= \left(\begin{pmatrix}\mathbf{I} & -\mathbf{I} & -\mathbf{I} & \mathbf{I}\end{pmatrix}\right)^{\mathsf{T}} \end{aligned} \tag{3.43}$$

and

$$\underline{b} = \begin{pmatrix}0 & 0 & -t & -t\end{pmatrix}, \tag{3.44}$$

respectively. It can be noted that representing the clipping function in convolutional form requires *four* times more channels than the original input signal.

Network depth

The relationship between network depth and low-rank approximation. It should be noted that one of the key elements of CNNs is the network depth, which we address in this section. To illustrate the effect of network depth, assume an arbitrary N-layer encoding-decoding CNN, in which the encoding layers are defined by

$$\begin{aligned}\underline{\mathbf{E}}_0 =&(\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0)_+,\\ \underline{\mathbf{E}}_1 =&(\underline{\mathbf{E}}_0 \circledast \underline{\mathbf{K}}_1 + \underline{b}_1)_+,\\ \underline{\mathbf{E}}_2 =&(\underline{\mathbf{E}}_1 \circledast \underline{\mathbf{K}}_2 + \underline{b}_2)_+,\\ &\vdots\\ \underline{\mathbf{E}}_{N-1} =&(\underline{\mathbf{E}}_{N-2} \circledast \underline{\mathbf{K}}_{N-1} + \underline{b}_{N-1})_+,\end{aligned} \tag{3.45}$$

$$\boxed{\underline{\mathbf{E}}_n = (\underline{\mathbf{E}}_{n-1} \circledast \underline{\mathbf{K}}_n + \underline{b}_n)_+} \quad . \tag{3.46}$$

Here, $\underline{\mathbf{E}}_n$ represents the encoded signal at the n-th decomposition level, while $\underline{\mathbf{K}}_n$, $\underline{b}_n$ are the convolution weights and biases for the n-th encoding layer, respectively. As addressed in Sections 3.3.3-A and 3.4.1, the role of the ReLU activations is to enforce sparsity and non-negativity, which can be interpreted as the process of suppressing non-informative bases in the low-rank approximation algorithm. Consequently, every encoded signal $\underline{\mathbf{E}}_n$, is an encoded *sparsified* version of the signal $\underline{\mathbf{E}}_{n-1}$. In order to recover the signal, we apply the decoder part of the CNN, given by

$$\begin{aligned}\tilde{\underline{\mathbf{E}}}_{N-1} =&(\underline{\mathbf{E}}_{N-1} \circledast \tilde{\underline{\mathbf{K}}}_{N-1}^{\mathsf{T}} + \tilde{\underline{b}}_{N-1})_+,\\ &\vdots\\ \tilde{\mathbf{E}}_1 =&(\tilde{\underline{\mathbf{E}}}_2 \circledast \tilde{\underline{\mathbf{K}}}_2^{\mathsf{T}} + \tilde{\underline{b}}_2)_+,\\ \tilde{\underline{\mathbf{E}}}_0 =&(\tilde{\underline{\mathbf{E}}}_1 \circledast \tilde{\underline{\mathbf{K}}}_1^{\mathsf{T}} + \tilde{\underline{b}}_1)_+,\\ \hat{\mathbf{x}} =&(\tilde{\underline{\mathbf{E}}}_0 \circledast \tilde{\underline{\mathbf{K}}}_0^{\mathsf{T}} + \tilde{\underline{b}}_0)_+,\end{aligned} \tag{3.47}$$

$$\boxed{\tilde{\underline{\mathbf{E}}}_{n-1} = (\tilde{\underline{\mathbf{E}}}_n \circledast \tilde{\underline{\mathbf{K}}}_n^{\mathsf{T}} + \tilde{\underline{b}}_n)_+} \quad . \tag{3.48}$$

Here, $\hat{\mathbf{x}}$ is the low-rank estimate/denoised version of the input signal $\mathbf{y}$, while $\tilde{\underline{\mathbf{E}}}_n$, $\tilde{\underline{\mathbf{K}}}_n^{\mathsf{T}}$, $\tilde{\underline{b}}_n$ are the decoded signal components at the n-th composition level and the decoder convolution weights and biases for the n-th layer, respectively. In Eq. (3.48) every decoded signal $\tilde{\underline{\mathbf{E}}}_n$ is the low-rank estimate of the encoded layer $\underline{\mathbf{E}}_{(n-1)}$. It should be noted that the activation of each of the decoder layers $(\cdot + \tilde{\underline{b}}_n)_+$ can further enforce sparsity on the low-rank estimates $\tilde{\underline{\mathbf{E}}}_{(n-1)}$.

Summary. In conclusion, the mathematical formulation of deep networks is analogous to a recursive data-driven low-rank approximation, where the input to the successive encoding-decoding pairs is the low-rank approximated encoded signal generated by the encoder of the previous level. Still, just as mentioned in the text box *Fitting low-rank approximation in ReLU CNNs*, low-rank approximation algorithms and CNNs are similar in terms of mathematical formulation, but we cannot ensure that the values obtained during training for the encoding, decoding filters and their biases have the properties needed to ensure that a CNN is an exact recursive data-driven low-rank approximation. For example, it is possible that the filters of the encoder and decoder do not reconstruct the signal perfectly, because this may not be necessary to reduce the loss function used to optimize the network.

Is it possible to impose a tighter relation between low-rank approximation and CNNs? In specific applications where signal preservation and interpretability is required (e.g. medical imaging) it is desirable that the operation of CNNs is closer to the low-rank approximation description. In order to achieve this, the CNNs embedded in frameworks such as the convolutional analysis operator [102] and FISTA-Net [111] explicitly train the filters $\underline{\mathbf{K}}_n$ and $\underline{\tilde{\mathbf{K}}}_n$ to have properties such as perfect signal reconstruction and sparsity. By enforcing these characteristics, the mathematical descriptions of the low-rank behavior and of CNNs are more similar and the models become inherently more interpretable and predictable on their operation.

3.4.4 Additional links between encoding-decoding CNNs and existing signal processing techniques

Up to this moment, it has been assumed that the operation of the encoding and decoding convolution filters are limited to map the input image to a sparse representation and to reconstruct it (i.e. $\underline{\mathbf{K}}$ and $\underline{\tilde{\mathbf{K}}}^{\intercal}$ comply with $\underline{\mathbf{K}} \circledast \underline{\tilde{\mathbf{K}}}^{\intercal} = \mathbf{I} \cdot c$). Still, it is possible that –besides decomposition and synthesis tasks– the encoding-decoding structure also filters/colors the signal in a way that improves the image estimates. It should be noted that this implies that the perfect reconstruction encoding-decoding structure is no longer preserved. For example, considering the following linear encoding-decoding structure

$$\hat{\mathbf{y}} = \mathbf{x} \circledast \underline{\mathbf{K}} \circledast \underline{\tilde{\mathbf{K}}}^{\intercal}, \tag{3.49}$$

which can be reduced to

$$\hat{\mathbf{y}} = \mathbf{x} * \mathbf{k}. \tag{3.50}$$

Here, $\mathbf{k} = \underline{\mathbf{K}} \circledast \underline{\tilde{\mathbf{K}}}^{\intercal}$ is optimized to reduce the distance between $\mathbf{x} * \mathbf{k}$ and the ground truth $\mathbf{y}$. Consequently, the equivalent filter $\mathbf{k}$ can be considered as being a *Wiener filter*. It should be noted that this text is not the first in addressing the potential Wiener-like behavior of a CNN. For example, Mohan *et al.* [104] suggested that by eliminating the bias of the convolution layers, the CNN could behave more akin to the Wiener filter and to be able to generalize better to unseen noise levels. It should be noted that by doing so, the CNN can also behave akin to the switching behavior described by the theory of deep convolutional framelets, which can is described by the equation

$$(z)_+ = \begin{cases} z, & \text{if } z \geq 0, \\ 0 & \text{if } z < 0, \end{cases}, \tag{3.51}$$

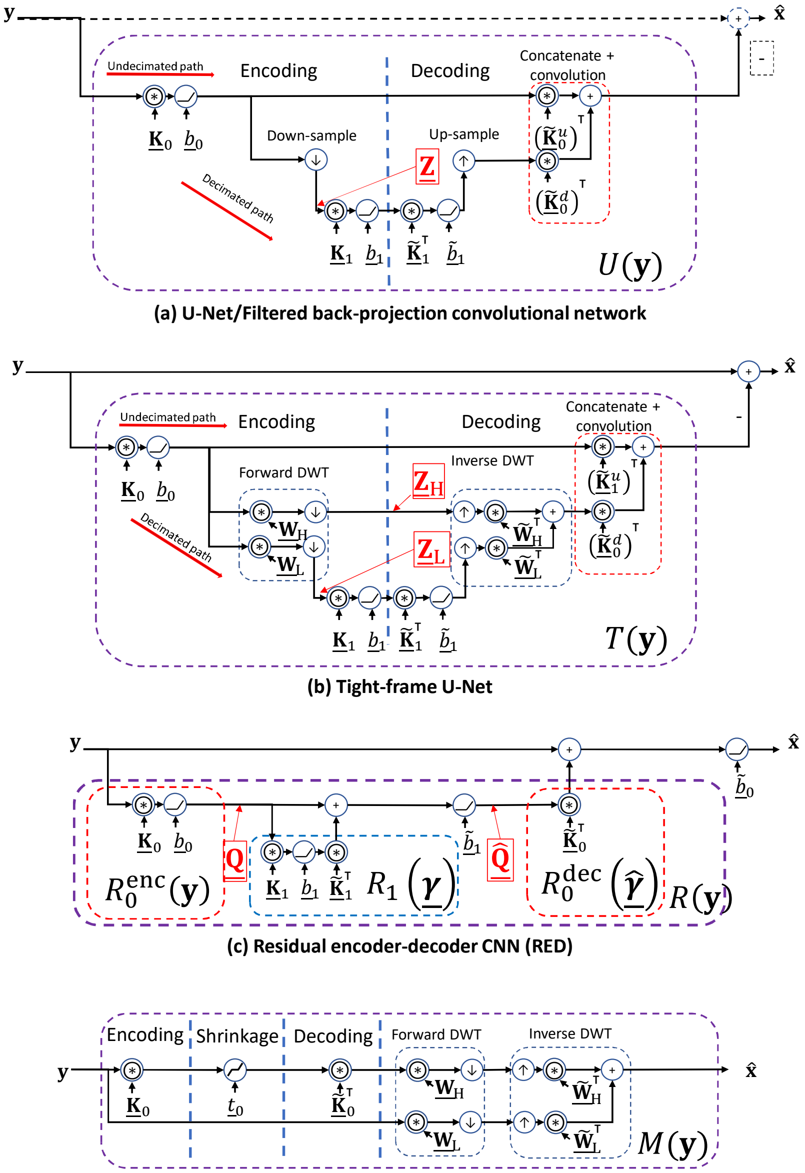
where z is a pixel which belongs to the signal $\mathbf{z} = \mathbf{x} * \mathbf{k}$. It can be observed that in contrast with the low-rank behavior described in Section 3.4.1, in this case the switching behavior is only dependent on the correlation between signal $\mathbf{x}$ and the filter $\mathbf{k}$. Consequently, if the value of z is positive, its value is preserved. On the contrary, if the correlation between $\mathbf{x}$ and $\mathbf{k}$ is negative, then the value of z is cancelled. Consequently, the noise reduction becomes independent/invariant of the noise level. It should be noted that this effect can be considered a non-linear extension of the so-called signal annihilation filters [112].

It should be noticed that besides the low-rank approximation interpretation of ReLU-based CNNs, additional links to other techniques can be derived. For example, the decomposition and synthesis provided by the encoding-decoding structure is also akin to the non-negative matrix factorization (NMF) [113], in which a signal is factorized as a weighted sum of positive bases. In this conception, the feature maps are the bases, which are constrained to be positive by the ReLU function. Furthermore, an additional interpretation of encoding-decoding CNNs can be obtained by analyzing them from a low-dimensional manifold representation perspective [100]. Here, the convolution layers of CNNs are interpreted as two operations. On one hand, they can provide a Hankel representation. On the other hand, they provide a bottleneck which reduces the dimensionality of the manifold of image patches. It should be noticed that the Hankel-like structure that is attributed to the convolution layers of CNNs, has also been noticed by the theory of the deep convolutional framelets [40]. A final connection with signal processing and CNNs is the variational formulation combined with kernel-based methods [105], [114], [115].

3.5 Analysis of relevant designs

In order to demonstrate the application of the principles summarized in Sections 3.3 and 3.4.3, this section analyzes relevant designs of ReLU and shrinkage CNNs. The analyses focus on three main aspects, which are: (1) the overall descriptions of the network architecture, (2) the signal propagation characteristics provided by the convolutional layers of the encoder and decoder sub-networks, and (3) the number operations $\mathcal{O}(\cdot)$ executed by the trainable parts of the network, since this will give insight on the computational requirements to execute each network and its overall complexity.

The signal reconstruction analysis provides a theoretical indication that a given CNN design can propagate *any* arbitrary signal when considering the use of ideal filters (i.e. they provide perfect reconstruction and are maximally sparse). In other words, for a fixed network architecture, there exists a selection of parameters (weights and biases) that make the neural network equal to the identity function. This result is important, because a design that cannot propagate arbitrary signals in ideal conditions,



(a) U-Net/Filtered back-projection convolutional network

(b) Tight-frame U-Net

(c) Residual encoder-decoder CNN (RED)

(d) Multi-scale sparse coding network (MSCN)

Figure 3.9. Simplified structure of encoding-decoding ReLU CNNs. The displayed networks are the U-Net/filtered back-projection network, the tight-frame U-Net, the residual encoder-decoder (RED) CNN and finally, the multi-scale sparse-coding network. Note that for all the designs, the encoding-decoding structures are indicated by dashed blocks. It should be beared in mind that the drawings are simplified, they do not contain normalization layers, are shallow, commonly appearing dual convolutions are drawn as one layer.

will potentially distort the signals that propagate through it *by design*. Consequently, this cannot be fixed by training with large datasets and/or with the application of any special loss term. In order to understand better the signal reconstruction analysis, we provide a brief example, where it is a non-residual CNN $G(\cdot)$, where we propagate a noiseless signal $\mathbf{x}$ contaminated with noise $\boldsymbol{\eta}$, so that

$$\mathbf{x} \approx G(\mathbf{x} + \boldsymbol{\eta}). \tag{3.52}$$

Here, an ideal CNN allows to propagate *any* $\mathbf{x}$, while cancelling the noise component $\boldsymbol{\eta}$, irrespective of the content of $\mathbf{x}$. If we switch our focus to an ideal *residual CNN* $R(\cdot)$, it is possible to observe that

$$\hat{\mathbf{x}} \approx R(\mathbf{y}) = \mathbf{y} - G(\mathbf{y}). \tag{3.53}$$

Here, $G(\cdot)$ is the encoding-decoding section of the residual network $R(\cdot)$. Consequently, it is desirable that the network $G(\cdot)$ is able to propagate the noise $\boldsymbol{\eta}$, while suppressing the noiseless signal $\mathbf{x}$, which is equivalent to

$$\eta \approx G(\mathbf{x} + \boldsymbol{\eta}). \tag{3.54}$$

It should be noted that in both residual and non-residual cases, there are two behaviors. On one hand, there is a signal which the network decomposes and reconstructs (almost) perfectly, and on the other hand a signal is suppressed. The signal reconstruction analysis focuses on the signals that the network can propagate or reconstruct, rather than the signal cancellation behavior. In consequence, we focus on the linear part of $G(\cdot)$ (i.e. its convolution structure), of which, according to Section 3.4.1, we assume that it properly handles the decomposition and reconstruction of the signal within the CNN. It should be noted that the idealized model assumed here, is only considered for analysis purposes, since practical implementations do no guarantee that this exact behavior is factually obtained. The reader is referred to Section 3.4.4 and the text boxes *Fitting low-rank approximation in ReLU CNNs* and *Network depth*.

In order to test the perfect reconstruction in non-residual CNNs, we propose the following procedure. (1) We assume an *idealized* model $G(\cdot)$, where its convolution filters $\underline{\mathbf{K}}_n$ and $\underline{\tilde{\mathbf{K}}}_n$ comply with the phase-complementary tight framelet condition and where the biases and non-linearities suppress low-amplitude (and negative for ReLU activations) samples from the feature maps. (2) The biases/thresholds of ReLU/shrinkage CNNs are set to zero (or to infinity for clipping activations). It can be observed that this condition prevents the low-rank (or high-rank for residual models) approximation behavior of the idealized CNN. Under this circumstance, it should be possible to prove that the analyzed CNN can perfectly reconstruct *any* signal. (3) The last step involves simplifying the mathematical description of the resulting model of the previous point. The mathematical simplification of the model should lead to the identity function if the model complies with the perfect reconstruction property.

To conclude the explanation on the perfect reconstruction analysis, we provide two relevant considerations. First, it can be claimed that residual networks, such as the model $R(\mathbf{y}) = \mathbf{y} - G(\mathbf{y})$ discussed in Eq. (3.53), is able to reconstruct any signal when $G(\mathbf{y}) = 0$ for any $\mathbf{y} = \mathbf{x} + \boldsymbol{\eta}$. Still, this does not convey information about the behavior of the encoding-decoding network $G(\cdot)$, which should be able to perform perfect decomposition and reconstruction of the noise signal $\boldsymbol{\eta}$, as discussed in Eq. (3.53). To avoid this trivial solution, instead of analyzing the network $R(\cdot)$, the analysis described for non-residual models is applied to the encoding-decoding structure $G(\cdot)$, which means that the residual connection is excluded from the analysis.

The second concluding remark is that in order to distinguish the equations of the perfect signal reconstruction analysis from other models, we specify the analyzed designs of the perfect reconstruction models, in which the low-rank approximation behavior is avoided by setting the bias values to zero, with a special operator $\mathscr{P}\{\cdot\}$.

For the analyses regarding the total number of operations of the trainable parameters, it is assumed that the tensors $\underline{\mathbf{K}}_0$, $\tilde{\underline{\mathbf{K}}}_0$, $\tilde{\underline{\mathbf{K}}}^{\mathrm{u}}_0$, $\tilde{\underline{\mathbf{K}}}^{\mathrm{d}}_0$, $\underline{\mathbf{K}}_1$ and $\tilde{\underline{\mathbf{K}}}_1$ shown in Fig. 3.9 have dimensions $(1 \times C_0 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, $(1 \times C_0 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, $(1 \times C_0 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, $(C_0 \times 1 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, $(C_0 \times C_1 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, $(C_0 \times C_1 \times N_{\mathrm{f}} \times N_{\mathrm{f}})$, respectively. Here, C_0 and C_1 represent the number of channels after the first and second convolution layer, all the convolution filters are assumed to be square with $N_{\mathrm{f}} \times N_{\mathrm{f}}$ pixels. Furthermore, the input signal $\mathbf{x}$ has dimensions $(1 \times 1 \times N_{\mathrm{r}} \times N_{\mathrm{c}})$, where N_{r} and N_{c} denote the numbers of rows and columns of signal $\mathbf{x}$, respectively.

Limitations. The analyses shown for the different networks in this chapter have the following limitations. (1) The analyzed networks have only enough decomposition levels and convolution layers to understand their basic operation. The motivation for this simplification is to keep the analyses short and clear. Moreover, the same principles can be extended to deeper networks, because a deep CNN can be considered a recursive embedding of single-level models. (2) The normalization layers are not considered, because they are linear operators which provide mean shifts and amplitude scaling. Consequently, for analysis purposes it can be assumed that they are embedded in the convolution weights. (3) For every encoder convolution kernel, it is assumed that there is at least one decoder filter. (4) No co-adaptations between the filters of the encoder and decoder layers are considered.

The remainder of this section shows analyses of a selection of a few representative designs. Specifically, the chosen designs are the *U-Net* [8] and its residual counterpart the *filtered back-projection network* [50] [1], the *tight-frame U-Net* [2], the *residual encoder-decoder CNN* [37] [3], and finally, the *multi-scale sparse-coding network*. For reference, all the designs are portrayed in Fig. 3.9.

[1] Matlab implementation by their authors available at `https://github.com/panakino/FBPConvNet`

[2] Pytorch implementation available at `https://github.com/LuisAlbertZM/demo_LWFSN_TMI` and interactive demo available at IEEE's code ocean `https://codeocean.com/capsule/9027829/tree/v1`. The demo also includes as reference pytorch implementations of FBPConvNet and the tight-frame U-Net.

[3] Pytorch implementation by their authors available at `https://github.com/SSinyu/RED-CNN`

3.5.1 U-Net/filtered back-projection network

A U-Net – overview of the design

The first networks analyzed are the U-Net and filtered back-projection networks, both of which share the encoding-decoding structure $U(\cdot)$. However, they differ in the fact that the U-Net is non-residual, while the filtered back-projection network operates in residual configuration. Therefore, the estimate of the noiseless signal $\hat{\mathbf{x}}$ from the noisy observation $\mathbf{y}$ in the conventional U-Net is achieved by

$$\hat{\mathbf{x}} = U(\mathbf{y}), \tag{3.55}$$

whereas in the filtered back-projection network, $U(\cdot)$ is used in residual configuration, which is equivalent to

$$\hat{\mathbf{x}} = \mathbf{y} - U(\mathbf{y}). \tag{3.56}$$

If we now switch focus to the encoding-decoding structure of the U-Net $U(\mathbf{y})$, it can be shown that it is described by

$$U(\mathbf{y}) = U^{\mathrm{u}}(\mathbf{y}) + U^{\mathrm{d}}(\mathbf{y}), \tag{3.57}$$

where $U^{\mathrm{u}}(\mathbf{y})$ corresponds to the *undecimated path*, which is defined by

$$U^{\mathrm{u}}(\mathbf{y}) = \big(\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0\big)_+ \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{\mathrm{u}}\big)^{\mathsf{T}}, \tag{3.58}$$

while the decimated path is

$$U^{\mathrm{d}}(\mathbf{y}) = f_{(2\uparrow)}\Big(\big((\underline{\mathbf{Z}} \circledast \underline{\mathbf{K}}_1 + \underline{b}_1)_+ \circledast \underline{\tilde{\mathbf{K}}}_1^{\mathsf{T}} + \underline{\tilde{b}}_1\big)_+\Big) \circledast \underline{\tilde{\mathbf{W}}}_{\mathrm{L}}^{\mathsf{T}} \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{d}\big)^{\mathsf{T}}. \tag{3.59}$$

Here, signal $\underline{\mathbf{Z}}$ is defined by

$$\underline{\mathbf{Z}} = f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0)_+ \circledast \underline{\mathbf{W}}_{\mathrm{L}}\big). \tag{3.60}$$

Note the decimated path contains two nested encoding-decoding architectures, as observed by Jin *et al.* [50], who has acknowledged that the nested filtering structure is akin to the (learned) iterative soft thresholding algorithm [109], [110].

B U-Net – reconstruction analysis

To prove if the U-Net can perfectly reconstruct *any* signal, we assume that the biases are equal to zero, on this condition the network $\mathscr{P}\{U\}(\mathbf{y})$ is defined by

$$\mathscr{P}\{U\}(\mathbf{y}) = \mathscr{P}\{U^{\mathrm{u}}\}(\mathbf{y}) + \mathscr{P}\{U^{\mathrm{d}}\}(\mathbf{y}), \tag{3.61}$$

where sub-network $\mathscr{P}\{U^{\mathrm{u}}\}(\cdot)$ is defined by

$$\mathscr{P}\{U^{\mathrm{u}}\}(\mathbf{y}) = (\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{\mathrm{u}}\big)^{\mathsf{T}}. \tag{3.62}$$

Assuming that $(\underline{\mathbf{K}}_0, \tilde{\underline{\mathbf{K}}}_0^{\mathrm{u}})$ is a complementary-phase tight-framelet pair then $\mathscr{P}\{U^{\mathrm{u}}\}(\mathbf{y})$ is simplified to

$$\mathscr{P}\{U^{\mathrm{u}}\}(\mathbf{y}) = \mathbf{y} \cdot c_0. \tag{3.63}$$

Furthermore, the low-frequency path is

$$U^{\mathrm{d}}(\mathbf{y}) = f_{(2\uparrow)}\Big(\big((\underline{\mathbf{Z}} \circledast \underline{\mathbf{K}}_1)_+ \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal}\big)_+\Big) \circledast \tilde{\underline{\mathbf{W}}}_{\mathrm{L}}^{\intercal} \circledast \big(\tilde{\underline{\mathbf{K}}}_0^{\mathrm{d}}\big)^{\intercal}\,, \tag{3.64}$$

where $\underline{\mathbf{Z}}$ is defined by

$$\underline{\mathbf{Z}} = f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \underline{\mathbf{W}}_{\mathrm{L}}\big). \tag{3.65}$$

If $\underline{\mathbf{K}}_1$ is a phase-complementary tight frame, we know that $(\underline{\mathbf{Z}} \circledast \underline{\mathbf{K}}_1)_+ \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} = \underline{\mathbf{Z}} \cdot c_1$. Consequently, Eq. (3.64) becomes

$$\mathscr{P}\{U^{\mathrm{d}}\}(\mathbf{y}) = f_{(2\uparrow)}(f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \underline{\mathbf{W}}_{\mathrm{L}}\big)) \circledast \tilde{\underline{\mathbf{W}}}_{\mathrm{L}}^{\intercal} \circledast \big(\tilde{\underline{\mathbf{K}}}_0^{\mathrm{d}}\big)^{\intercal} \cdot c_1. \tag{3.66}$$

Here, it can be noticed that if $\underline{\mathbf{K}}_0$ is a phase-complementary tight framelet, then $\mathscr{P}\{U^{\mathrm{d}}\}(\mathbf{y})$ approximates a low-pass version of $\mathbf{y}$, or equivalently

$$\mathscr{P}\{U^{\mathrm{d}}\}(\mathbf{y}) \approx \mathbf{y} \circledast \underline{\mathcal{W}}_{\mathrm{L}} \cdot c_1, \tag{3.67}$$

where $\mathcal{W}_{\mathrm{L}}$ is a low-pass filter. Finally, substituting Eq. (3.63) and Eq. (3.67) in Eq. (3.61) results in

$$\mathscr{P}\{U\}(\mathbf{y}) \approx \mathbf{y} \circledast (\mathbf{I} \cdot c_0 + \underline{\mathcal{W}}_{\mathrm{L}} \cdot c_1). \tag{3.68}$$

This result proves that the design of the U-Net cannot evenly reconstruct all the frequencies of $\mathbf{y}$ unless $c_1 = 0$, in which case the whole low-frequency branch of the network is ignored. Note that this limitation is inherent to its design and cannot be circumvented by training with large datasets and/or with any loss function.

C U-Net – number of operations

It can be noted that encoding filter $\underline{\mathbf{K}}_0$ convolves $\mathbf{x}$ at its original resolution and maps it to a tensor with C_0 channels. Therefore, the number of operations $\mathcal{O}(\cdot)$ for kernel $\underline{\mathbf{K}}_0$ is $\mathcal{O}(\underline{\mathbf{K}}_0) = C_0 \cdot N_{\mathrm{r}} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{f}}^2$ [FLOPS] (floating-point operations). Conversely, due to the symmetry between encoder and decoder filters, $\mathcal{O}(\tilde{\underline{\mathbf{K}}}_0^{\mathrm{u}}) = \mathcal{O}(\tilde{\underline{\mathbf{K}}}_0^{\mathrm{d}}) = \mathcal{O}(\underline{\mathbf{K}}_0)$. Furthermore, for this design, filter $\underline{\mathbf{K}}_1$ processes the signal encoded by $\underline{\mathbf{K}}_0$, which is down-sampled by a factor of one half for and maps it from C_0 to C_1 channels, this results in the estimated operation cost $\mathcal{O}(\underline{\mathbf{K}}_1) = \mathcal{O}(\tilde{\underline{\mathbf{K}}}_1) = C_0 \cdot C_1 \cdot N_{\mathrm{r}} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{f}}^2 \cdot (2)^{-2}$ [FLOPs]. Finally, adding the contributions of filters $\underline{\mathbf{K}}_0$, $\tilde{\underline{\mathbf{K}}}_0^{\mathrm{u}}$, $\tilde{\underline{\mathbf{K}}}_0^{\mathrm{d}}$, $\underline{\mathbf{K}}_1$ and $\tilde{\underline{\mathbf{K}}}_1$ results in

$$\mathcal{O}(U) = (3 + 2^{-1} \cdot C_1) \cdot C_0 \cdot N_{\mathrm{r}} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{f}}^2 \text{ [FLOPS]} \tag{3.69}$$

D U-Net – Concluding remarks

The U-Net/FBPConvNet is a flexible multi-resolution architecture. As it has been shown, the pooling structure of this CNN may be still sub-optimal for noise reduction applications because this configuration does not allow to recover the frequency information of the signal evenly. This has been noted and fixed by Han and Ye [36], who introduced the so-called tight-frame U-Net, where the down/up-sampling structure is replaced by the discrete wavelet transform and its inverse.

3.5.2 Tight-frame U-Net

A Tight-frame-U-Net – Overview of the design

In order to avoid the low-frequency over-emphasis caused by the U-Net, Han *et al.* proposed the so-called tight-frame U-Net [36]. In this network, the up- and down-sampling layers of the U-Net are replaced by the discrete wavelet transform and its inverse. This simple modification eliminates the blurring associated to the conventional U-Net.

The tight-frame U-Net depicted in Fig. 3.9 (b) is used in residual configuration. Therefore, its estimates are obtained by

$$\hat{\mathbf{x}} = \mathbf{y} - T(\mathbf{y}). \tag{3.70}$$

Here, $T(\cdot)$ represents the encoding-decoding structure of the tight-frame U-Net, described by

$$T(\mathbf{y}) = T^{\mathrm{u}}(\mathbf{y}) + T^{\mathrm{d}}(\mathbf{y}). \tag{3.71}$$

In the above equation, $T^{\mathrm{u}}(\mathbf{y})$ is the undecimated path, defined by

$$T^{\mathrm{u}}(\mathbf{y}) = \big(\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0\big)_+ \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{u}\big)^{\intercal}, \tag{3.72}$$

which is equivalent to the undecimated path of the conventional U-Net. In contrast with the conventional U-Net, the tight-frame U-Net uses the discrete wavelet transform as down- and up-sampling layer. This modification bifurcates the signal in low- and high-frequency paths, which is equivalent to

$$T^{\mathrm{d}}(\mathbf{y}) = T^{\mathrm{d}}_{\mathrm{H}}(\mathbf{y}) + T^{\mathrm{d}}_{\mathrm{L}}(\mathbf{y}), \tag{3.73}$$

where the simplest, high-frequency path $T^{\mathrm{d}}_{\mathrm{H}}(\mathbf{y})$ is

$$T^{\mathrm{d}}_{\mathrm{H}}(\mathbf{y}) = f_{(2\uparrow)}\big(\underline{\mathbf{Z}}_{\mathrm{H}}\big) \circledast \underline{\tilde{\mathbf{W}}}_{\mathrm{H}}^{\intercal} \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{\mathrm{d}}\big)^{\intercal}, \tag{3.74}$$

and $\underline{\mathbf{Z}}_{\mathrm{H}}$ is

$$\underline{\mathbf{Z}}_{\mathrm{H}} = f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0)_+ \circledast \underline{\mathbf{W}}_{\mathrm{H}}\big). \tag{3.75}$$

Furthermore, the more elaborated low-frequency path has an extra convolution compared with the high-frequency path, is described by

$$T^{\mathrm{d}}_{\mathrm{L}}(\mathbf{y}) = f_{(2\uparrow)}\Big(\big((\underline{\mathbf{Z}}_{\mathrm{L}} \circledast \underline{\mathbf{K}}_1 + \underline{b}_1\big)_+ \circledast \underline{\tilde{\mathbf{K}}}_1^{\intercal} + \underline{\tilde{b}}_1\big)_+\Big) \circledast \underline{\tilde{\mathbf{W}}}_{\mathrm{L}}^{\intercal} \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{d}\big)^{\intercal}. \tag{3.76}$$

In this case, the signal $\underline{\mathbf{Z}}_\mathrm{L}$ is in the low-frequency path is defined by

$$\underline{\mathbf{Z}}_\mathrm{L} = f_{(2\downarrow)}\big(\mathbf{y} \circledast \underline{\mathbf{K}}_0 + \underline{b}_0\big)_+ \circledast \underline{\mathbf{W}}_\mathrm{L}. \tag{3.77}$$

Note that the filtering structure is very similar to the U-Net/filtered back-projection network, but the design contains an additional high-frequency skip connection, which is extracted with filter $\underline{\mathbf{W}}_\mathrm{H}$.

B Tight-frame U-Net – signal reconstruction analysis

When analyzing the signal reconstruction on the tight-frame U-Net, the equation describing the (un)decimated paths Eq. (3.71) becomes

$$\mathscr{P}\{T\}(\mathbf{y}) = \mathscr{P}\{T^\mathrm{u}\}(\mathbf{y}) + \mathscr{P}\{T^\mathrm{d}\}(\mathbf{y}), \tag{3.78}$$

where the undecimated signal path described by Eq. (3.72) is

$$\mathscr{P}\{T^\mathrm{u}\}(\mathbf{y}) = (\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \big(\tilde{\underline{\mathbf{K}}}_0^u\big)^\intercal. \tag{3.79}$$

It is assumed that the filter pair $(\underline{\mathbf{K}}_0, \tilde{\underline{\mathbf{K}}}_0^\mathrm{u})$ complies with the phase-complementary tight-framelet property. This reduces $\mathscr{P}\{T^\mathrm{u}\}(\mathbf{y})$ to

$$\mathscr{P}\{T^\mathrm{u}(\mathbf{y})\} = \mathbf{y} \cdot c_u. \tag{3.80}$$

Furthermore, the decimated path $\mathscr{P}\{T^\mathrm{d}\}(\cdot)$ is defined by

$$\mathscr{P}\{T^\mathrm{d}\}(\mathbf{y}) = \mathscr{P}\{T^\mathrm{d}_\mathrm{H}\}(\mathbf{y}) + \mathscr{P}\{T^\mathrm{d}_\mathrm{L}\}(\mathbf{y}), \tag{3.81}$$

where the high-frequency branch $\mathscr{P}\{T^\mathrm{d}_\mathrm{H}\}$ is

$$\mathscr{P}\{T^\mathrm{d}_\mathrm{H}\}(\mathbf{y}) = f_{(2\uparrow)}\big(f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \underline{\mathbf{W}}_\mathrm{H}\big)\big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{H}^\intercal \circledast \big(\tilde{\underline{\mathbf{K}}}_0^\mathrm{d}\big)^\intercal. \tag{3.82}$$

Similarly, the low-frequency branch $\mathscr{P}\{T^\mathrm{d}_\mathrm{L}\}$ of Eq. (3.81) is

$$\mathscr{P}\{T^\mathrm{d}_\mathrm{L}\}(\mathbf{y}) = f_{(2\uparrow)}\Big(\big((\underline{\mathbf{Z}}_\mathrm{L} \circledast \underline{\mathbf{K}}_1)_+ \circledast \tilde{\underline{\mathbf{K}}}_1^\intercal\big)_+\Big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{L}^\intercal \circledast \big(\tilde{\underline{\mathbf{K}}}_0^d\big)^\intercal, \tag{3.83}$$

where signal $\mathscr{P}\{\underline{\mathbf{Z}}_\mathrm{L}\}$ is defined by

$$\mathscr{P}\{\underline{\mathbf{Z}}_\mathrm{L}\} = f_{(2\downarrow)}\Big((\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \underline{\mathbf{W}}_\mathrm{L}\Big). \tag{3.84}$$

Assuming $(\underline{\mathbf{K}}_1, \tilde{\underline{\mathbf{K}}}_1)$ is a complementary-phase tight-framelet pair, Eq. (3.83) is simplified to

$$\mathscr{P}\{T^\mathrm{d}_\mathrm{L}\}(\mathbf{y}) = f_{(2\uparrow)}\Big((\underline{\mathbf{Z}}_\mathrm{L})_+\Big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{L}^\intercal \circledast \big(\tilde{\underline{\mathbf{K}}}_0^d\big)^\intercal. \tag{3.85}$$

If $\underline{\mathbf{Z}}_\mathrm{L} \geq 0$, we obtain

$$\mathscr{P}\{T^\mathrm{d}_\mathrm{L}\}(\mathbf{y}) = f_{(2\uparrow)}\big(f_{(2\downarrow)}\big(\mathbf{y} \circledast \underline{\mathbf{K}}_0 \circledast \underline{\mathbf{W}}_\mathrm{L}\big)\big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{L}^\intercal \circledast \big(\tilde{\underline{\mathbf{K}}}_0^d\big)^\intercal. \tag{3.86}$$

When adding the high- and low-frequency branches described in Eqs. (3.82) and (3.86) in Eq. (3.81), the result becomes

$$\mathscr{P}\{T^{\mathrm{d}}\}(\mathbf{y}) = f_{(2\uparrow)}\big(f_{(2\downarrow)}\big((\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \underline{\mathbf{W}}\big)\big) \circledast \underline{\tilde{\mathbf{W}}}^{\mathsf{T}} \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{d}\big)^{\mathsf{T}}. \tag{3.87}$$

Furthermore, for the discrete wavelet transform, any signal $\underline{\mathbf{Q}}$ can be perfectly reconstructed by $\underline{\mathbf{Q}} = f_{(2\uparrow)}\big(f_{(2\downarrow)}\big(\underline{\mathbf{Q}} \circledast \mathbf{W}\big)\big) \circledast \underline{\tilde{\mathbf{W}}}^{\mathsf{T}}$. Consequently, Eq. (3.87) can be simplified to

$$\mathscr{P}\{T^{\mathrm{d}}\}(\mathbf{y}) = (\mathbf{y} \circledast \underline{\mathbf{K}}_0)_+ \circledast \big(\underline{\tilde{\mathbf{K}}}_0^{d}\big)^{\mathsf{T}}, \tag{3.88}$$

and then further reduced to

$$\mathscr{P}\{T^{\mathrm{d}}\}(\mathbf{y}) = \mathbf{y} \cdot c_{\mathrm{d}}. \tag{3.89}$$

Inserting Eq. (3.89) and Eq. (3.80) in Eq. (3.78), leads to

$$\mathscr{P}\{T\}(\mathbf{y}) = \mathbf{y} \cdot c_{\mathrm{u}} + \mathbf{y} \cdot c_{\mathrm{d}}. \tag{3.90}$$

If $c_{\mathrm{u}} + c_{\mathrm{d}} = 1$, the expression becomes

$$\mathscr{P}\{T\}(\mathbf{y}) = \mathbf{y}. \tag{3.91}$$

Eq. (3.91) proves that the tight-frame U-Net can perfectly reconstruct any signal. This finding proves that the additional high-frequency skip-connection introduced by Han *et al.* improves the signal propagation characteristics when compared to the conventional U-Net.

C Tight-frame U-Net – number of operations

In the case of the tight-frame U-Net, it can be observed that the convolutional structure is identical to the original U-Net, therefore the number of operations executed by the convolutional layers is also equal to

$$\mathcal{O}(T) = (3 + 2^{-1} \cdot C_1) \cdot C_0 \cdot N_{\mathrm{r}} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{f}}^2 \text{ [FLOPS]}. \tag{3.92}$$

D Tight frame U-Net – Concluding remarks

The tight frame U-Net is a multi-resolution design which overcomes the limitations of the conventional U-Net by means of using an additional high-pass skip connection.

In order to allow to reconstruct any arbitrary signal through the convolutional structure of the tight frame U-Net, multiple conditions should be satisfied. First, the encoding-decoding filter pairs $(\underline{\mathbf{K}}_0, \underline{\tilde{\mathbf{K}}}_0^{d})$, $(\underline{\mathbf{K}}_0, \underline{\tilde{\mathbf{K}}}_0^{u})$ and $(\underline{\mathbf{K}}_1, \underline{\tilde{\mathbf{K}}}_1)$ should be complementary-phase tight framelets. In addition, filters $(\underline{\mathbf{K}}_1, \underline{\tilde{\mathbf{K}}}_1^{u})$ also should comply with the condition $\underline{\mathbf{Z}}_{\mathrm{L}} \geq 0$.

Furthermore, for simplification it should be noted that the convolution kernel $\underline{\mathbf{K}}_0$ has two inverse convolution kernels ($\underline{\tilde{\mathbf{K}}}_0^{d}$ and $\underline{\tilde{\mathbf{K}}}_0^{u}$). Therefore, the skip connection that is convolved with the kernel $\underline{\tilde{\mathbf{K}}}_0^{u}$ could potentially be eliminated to save parameters in the decoder.

3.5.3 Residual encoder-decoder CNN

A Residual encoder-decoder CNN – overview of the design

The residual encoder-decoder CNN shown in Fig. 3.9 (c) consists of nested single-layer residual encoding-decoding networks. For example, in the network showcased in Fig. 3.9 (c) it is visible that network $R_1(\cdot)$ is nested into $R_0(\cdot)$. Furthermore, for this case the image estimate is given by

$$\hat{\mathbf{x}} = \left(\mathbf{y} + R_0(\mathbf{y}) + \underline{\tilde{b}}_0\right)_+, \tag{3.93}$$

in which $R_0(\cdot)$ is the outer residual network and $\underline{\tilde{b}}_0$ is the bias for the output layer. Note that the ReLU placed at the output layer intrinsically assumes that the estimated signal $\hat{\mathbf{x}}$ is positive.

From Eq. (3.93), the output of the sub-network $R_0(\cdot)$ is defined by

$$\underline{\mathbf{Z}} = R_0^{\text{dec}}(\underline{\hat{\mathbf{Q}}}). \tag{3.94}$$

Here, the decoder $R_0^{\text{dec}}(\cdot)$ is defined by

$$R_0^{\text{dec}}(\underline{\hat{\mathbf{Q}}}) = \underline{\hat{\mathbf{Q}}} \circledast \underline{\tilde{\mathbf{K}}}_0^{\intercal}. \tag{3.95}$$

In the above, $\underline{\hat{\mathbf{Q}}}$ is the noiseless estimate of the intermediate signal $\underline{\mathbf{Q}}$ (see Fig. 3.9 (c)) and it is defined by

$$\underline{\hat{\mathbf{Q}}} = (\underline{\mathbf{Q}} + R_1(\underline{\mathbf{Q}}) + \underline{\tilde{b}}_1)_+, \tag{3.96}$$

where the network $R_1(\cdot)$ is

$$R_1(\underline{\mathbf{Q}}) = \left(\underline{\mathbf{Q}} \circledast \underline{\mathbf{K}}_1 + \underline{b}_1\right)_+ \circledast \underline{\tilde{\mathbf{K}}}_1^{\intercal}. \tag{3.97}$$

Furthermore, $\underline{\mathbf{Q}}$ represents the signal encoded by $R_0(\cdot)$, or equivalently

$$\underline{\mathbf{Q}} = R_0^{\text{enc}}(\mathbf{y}), \tag{3.98}$$

where $R_0^{\text{enc}}(\cdot)$ is defined by

$$R_0^{\text{enc}}(\mathbf{y}) = \mathbf{y} \circledast \underline{\mathbf{K}}_0. \tag{3.99}$$

B Residual encoder-decoder CNN – reconstruction analysis

As mentioned earlier, the residual encoder-decoder CNN is composed by nested residual blocks, which are independently analyzed to study the reconstruction characteristics of this network. First, for block $R_1(\cdot)$, the linear representation is given by

$$\mathscr{P}\{R_1\}(\underline{\mathbf{Q}}) = (\underline{\mathbf{Q}} \circledast \underline{\mathbf{K}}_1)_+ \circledast \underline{\tilde{\mathbf{K}}}_1^{\intercal}. \tag{3.100}$$

Under complementary-phase tight-frame assumptions for the pair $(\underline{\mathbf{K}}_1, \underline{\tilde{\mathbf{K}}}_1)$, Eq. (3.100) reduces to

$$\mathscr{P}\{R_1\}(\underline{\mathbf{Q}}) = \underline{\mathbf{Q}}, \tag{3.101}$$

which shows that the encoder and decoder $R_1(\cdot)$ can approximately reconstruct any signal. When switching to R_0, it can be observed that the linear part is

$$\mathscr{P}\{R_0\}(\mathbf{y}) = \mathbf{y} \circledast (\underline{\mathbf{K}}_0 \circledast \underline{\mathbf{Q}})_+ \circledast \tilde{\underline{\mathbf{K}}}_0^{\mathsf{T}}. \tag{3.102}$$

Just as with $R_1(\cdot)$, it is assumed that the convolution kernels are tight-framelets. Therefore, Eq. (3.102) becomes

$$\mathscr{P}\{R_0\}(\mathbf{y}) = \mathbf{y}. \tag{3.103}$$

Consequently, $R_0(\cdot)$ and $R_1(\cdot)$ can reconstruct any arbitrary signal under complementary-phase tight-frame assumptions.

C Residual encoder-decoder CNN – number of operations

In this case, all the convolution layers operate at the original resolution of image $\mathbf{x}$. Therefore, the number of operations $O(\cdot)$ for kernel $\underline{\mathbf{K}}_0$ and $\tilde{\underline{\mathbf{K}}}_0$ is $O(\underline{\mathbf{K}}_0) = O(\tilde{\underline{\mathbf{K}}}_0) = C_0 \cdot N_\mathrm{r} \cdot N_\mathrm{c} \cdot N_\mathrm{f}^2$ [FLOPs], while $\underline{\mathbf{K}}_1$ and $\tilde{\underline{\mathbf{K}}}_1$ are requiring $O(\underline{\mathbf{K}}_1) = O(\tilde{\underline{\mathbf{K}}}_1) = C_0 \cdot C_1 \cdot N_\mathrm{r} \cdot N_\mathrm{c} \cdot N_\mathrm{f}^2$ [FLOPs]. By adding the contributions of both encoding-decoding pairs, the total operations for the residual encoder-decoder becomes

$$\mathcal{O}(R) = 2 \cdot (1 + C_1) \cdot C_0 \cdot N_\mathrm{r} \cdot N_\mathrm{c} \cdot N_\mathrm{f}^2 \text{ [FLOPS]}. \tag{3.104}$$

D Residual encoder-decoder CNN – Concluding remarks

The residual encoder-decoder network consists of a set of nested single-resolution residual encoding-decoding CNNs. The single-resolution design increases its computation cost with respect to multi-resolution designs such as the U-Net. In addition, it should be noted that the use of a ReLU as output layer of the encoder-decoder residual network forces the signal estimates to be positive, but this is not always convenient. For example in computed tomography imaging, it is common that images contain positive and negative values.

3.5.4 Multi-scale sparse coding network

A Multi-scale sparse coding network – description of architecture

The multi-scale sparse coding network is a multi-resolution architecture, in which the discrete wavelet transform is used as a mean for up/down-sampling. The filtering configuration is non-residual, therefore, the estimates are computed by

$$\hat{\mathbf{x}} = M(\mathbf{y}). \tag{3.105}$$

Here, the encoding-decoding structure $M(\cdot)$ is given by

$$\begin{aligned} M(\mathbf{y}) =& f_{(2\downarrow)}\big(f_{(2\uparrow)}\big(\tau^{\mathrm{Gar}}_{(\underline{t}_0 \cdot \sigma_\eta)}(\mathbf{y} \circledast \underline{\mathbf{K}}_0) \circledast \tilde{\underline{\mathbf{K}}}_0^{\mathsf{T}}\big) \circledast \underline{\mathbf{W}}_\mathrm{H}\big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{H}^{\mathsf{T}} + \\ & f_{(2\downarrow)}\big(f_{(2\uparrow)}(\mathbf{y}) \circledast \underline{\mathbf{W}}_\mathrm{L}\big) \circledast \tilde{\underline{\mathbf{W}}}_\mathrm{L}^{\mathsf{T}}, \end{aligned} \tag{3.106}$$

where σ_η is the standard deviation of the noise present in the image, which is also an input to the network. The noise parameter allows the network to adapt the threshold in function of the noise level. Furthermore, it can be noted that the filters of the discrete wavelet transform are not used as part of the noise reduction process, i.e. $\tau^{\text{Gar}}_{(\underline{t}_0 \cdot \sigma_\eta)}(\mathbf{y} \circledast \underline{\mathbf{K}}_0)$.

As concluding remark, we would like to mention that in deeper networks additional decomposition levels are placed in the low-frequency branch of the DWT, between the down- and up-sampling operations. In this condition, the additional encoding-decoding pairs operate at a lower resolution than the original input image. This makes the diagram less simplified than it looks, resulting in different considerations.

B Multi-scale sparse coding network – reconstruction analysis

When analyzing the perfect reconstruction of the model described by Eq. (3.106) we obtain

$$\begin{aligned}\mathscr{P}\{M\}(\mathbf{y}) =& f_{(2\downarrow)}\big(f_{(2\uparrow)}\big(\mathbf{y} \circledast \underline{\mathbf{K}}_0 \circledast \underline{\tilde{\mathbf{K}}}_0^{\intercal}\big) \circledast \underline{\mathbf{W}}_{\mathrm{H}}\big) \circledast \underline{\tilde{\mathbf{W}}}_{\mathrm{H}}^{\intercal} + \\ & f_{(2\downarrow)}\big(f_{(2\uparrow)}\big(\mathbf{y}\big) \circledast \underline{\mathbf{W}}_{\mathrm{L}}\big) \circledast \underline{\tilde{\mathbf{W}}}_{\mathrm{L}}^{\intercal}.\end{aligned} \tag{3.107}$$

Assuming that $\underline{\mathbf{K}}_0$ is a normalized framelet, i.e. $\underline{\mathbf{K}}_0 \circledast \underline{\tilde{\mathbf{K}}}_0^{\intercal} = \mathbf{I} \cdot c$, with $c = 1$, then Eq. (3.107) becomes

$$\mathscr{P}\{M\}(\mathbf{y}) = f_{(2\downarrow)}\big(f_{(2\uparrow)}\big(\mathbf{y}\big) \circledast \underline{\mathbf{W}}\big) \circledast \underline{\tilde{\mathbf{W}}}^{\intercal}, \tag{3.108}$$

which is further simplified to

$$\mathscr{P}\{M\}(\mathbf{y}) = \mathbf{y}. \tag{3.109}$$

As a result, the encoding-decoding structure of the multi-scale sparse coding network can potentially reconstruct any signal.

C Multi-scale sparse coding network – number of operations

The multi-scale sparse coding network contains a simpler convolution structure than the networks reviewed up to this moment. Therefore, for a single-level decomposition architecture, the total number of operations is

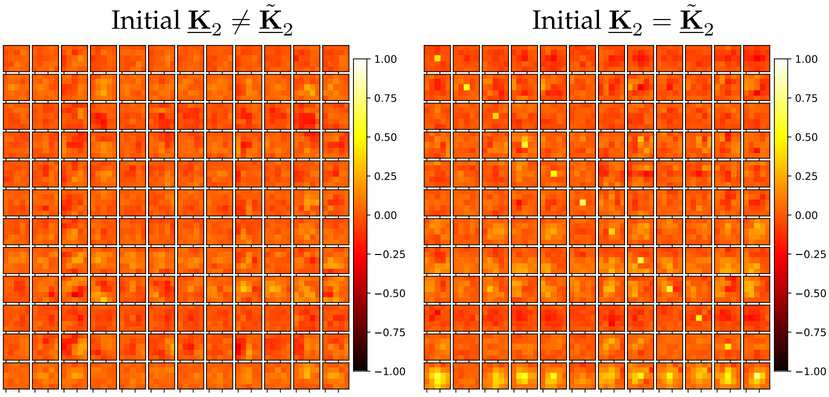
$$O(M) = 2 \cdot O(\underline{\tilde{\mathbf{K}}}_0) = 2 \cdot C_0 \cdot N_{\mathrm{r}} \cdot N_{\mathrm{c}} \cdot N_{\mathrm{f}}^2 \text{ [FLOPs]}. \tag{3.110}$$

D Multi-scale sparse coding network – Concluding remarks

As shown, the multi-scale sparse coding network is a shrinkage-based CNN, whose design allows for perfect signal reconstruction. Still, its design does not leverage the DWT to generate a sparser signal to further eliminate noise.

Design elements	No. of operations normalized by $(N_\mathrm{r} \cdot N_\mathrm{c} \cdot N_\mathrm{f}^2)$ [FLOPS]
U-Net	$(3 + 0.5 \cdot C_1) \cdot C_0$
Tight-frame U-Net	$(3 + 0.5 \cdot C_1) \cdot C_0$
Res. enc.-dec. net. (RED)	$2 \cdot (1 + C_1) \cdot C_0$
Multi-scale sparse coding net.	$2 \cdot C_0$

Table 3.1. Normalized number of computations performed by the analyzed CNNs


Initial $\underline{\mathbf{K}}_2 \neq \tilde{\underline{\mathbf{K}}}_2$

Initial $\underline{\mathbf{K}}_2 = \tilde{\underline{\mathbf{K}}}_2$

Figure 3.10. Phase-complementary tight-framelet test for the trained toy network, initialized with random weights. The left figure shows the product $(\underline{\mathbf{K}}_2)_+\tilde{\mathbf{K}}_2$, where the initialization of $\underline{\mathbf{K}}_2$ and $\tilde{\underline{\mathbf{K}}}_2$ is different. It can be seen that the pair $(\underline{\mathbf{K}}_2, \tilde{\underline{\mathbf{K}}}_2)$ does not comply with the complementary-phase framelet criterion of Eq. (3.111). This contrasts with the right result, which displays the result of the product $(\underline{\mathbf{K}}_2)_+\tilde{\underline{\mathbf{K}}}_2$, for the same CNN, but where the initial values of $\tilde{\underline{\mathbf{K}}}_2$ and $\underline{\mathbf{K}}_2$ are identical. For this initialization, the filters approximate the complementary-phase tight-framelet criterion.

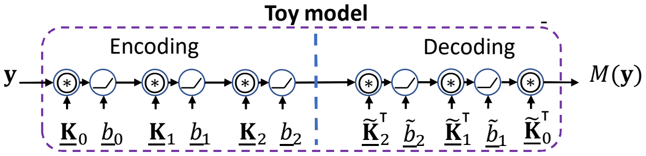

Toy model
Encoding
Decoding
$\mathbf{y}$
$M(\mathbf{y})$
$\underline{\mathbf{K}}_0$
$\underline{b}_0$
$\underline{\mathbf{K}}_1$
$\underline{b}_1$
$\underline{\mathbf{K}}_2$
$\underline{b}_2$
$\tilde{\underline{\mathbf{K}}}_2^\mathsf{T}$
$\tilde{\underline{b}}_2$
$\tilde{\underline{\mathbf{K}}}_1^\mathsf{T}$
$\tilde{\underline{b}}_1$
$\tilde{\underline{\mathbf{K}}}_0^\mathsf{T}$

Figure 3.11. Toy model used for experiment on the properties of the filters of a trained CNN. The dimensions for tensors $\mathbf{K}_0$, $\mathbf{K}_1$ and $\mathbf{K}_2$ are $(1 \times 6 \times 3 \times 3)$, $(6 \times 12 \times 3 \times 3)$ and $(12 \times 24 \times 3 \times 3)$. The network is symmetric and the filter dimensions for the decoder convolution kernels $\tilde{\mathbf{K}}_n$ are the same as there corresponding encoding kernel $\mathbf{K}_n$.

3.6 What happens in reality in trained models?

3.6.1 Filter properties and low-rank approximation

The assumption that the convolution filters of a CNN behave as (comple-mentary-phase) tight framelets is useful for analyzing the theoretical ability of a CNN to

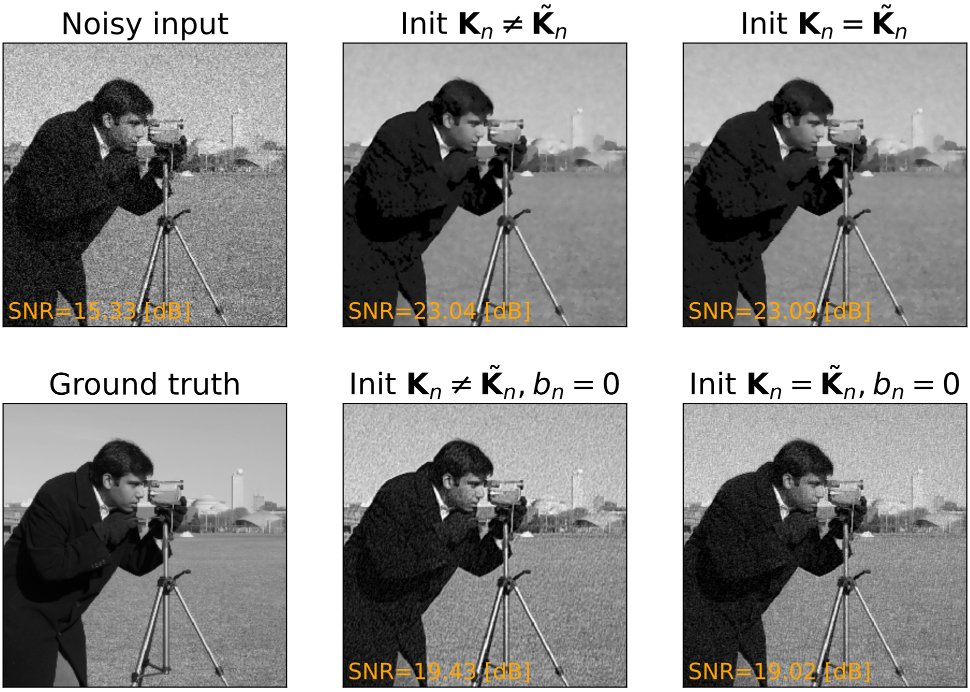

Noisy input
Init $\mathbf{K}_n \neq \tilde{\mathbf{K}}_n$
Init $\mathbf{K}_n = \tilde{\mathbf{K}}_n$
SNR=15.33 [dB]
SNR=23.04 [dB]
SNR=23.09 [dB]
Ground truth
Init $\mathbf{K}_n \neq \tilde{\mathbf{K}}_n, b_n = 0$
Init $\mathbf{K}_n = \tilde{\mathbf{K}}_n, b_n = 0$
SNR=19.43 [dB]
SNR=19.02 [dB]

Figure 3.12. Processed “cameraman" image for (in)dependently sampled initialization for the encoding and decoding filters. The top-left picture represents the noise-contaminated input ($\sigma_\eta = 0.1$) and the bottom-left, the noiseless reference. The middle-column images are the processed noisy image with the toy model trained with different initialization for its convolution filters, while the right-column images are processed with the model where the same initial values are used for the encoding and decoding filters. The top-middle and top-right images are nearly identical in terms of quality and SNR, so that initialization has no effect. The middle and bottom-right images are the same model presented that processed the middle and top-right figures, but where its bias is set to zero. As expected, the noise is partly reconstructed. Furthermore, it can be observed that in the model where the initial values are different, the reconstructed noisy input has a coarser nature and tends to be more visible.

propagate signals. However, it is difficult to prove that trained models comply with this assumption, because there are diverse elements affecting the optimization of the model, e.g. the initialization of the network, the data presented to the model, the optimization algorithm as well as its parameters. In addition, in real CNNs, there may be co-adaptation between the diverse CNN layers, which may prevent that the individual filters of the CNN behave as tight framelets, since the decomposition and filtering performed by one layer is not independent from the rest [116].

To test the behavior if the filters of a trained CNN can converge to complementary-phase tight framelets, at least on a simplified environment, we propose to train a toy model, as displayed in Fig. 3.11. If the trained filters of an encoder-decoder pair of the toy model ($\underline{\mathbf{K}}_l$, $\underline{\tilde{\mathbf{K}}}_l$), (where l denotes one of the decomposition levels) behave as

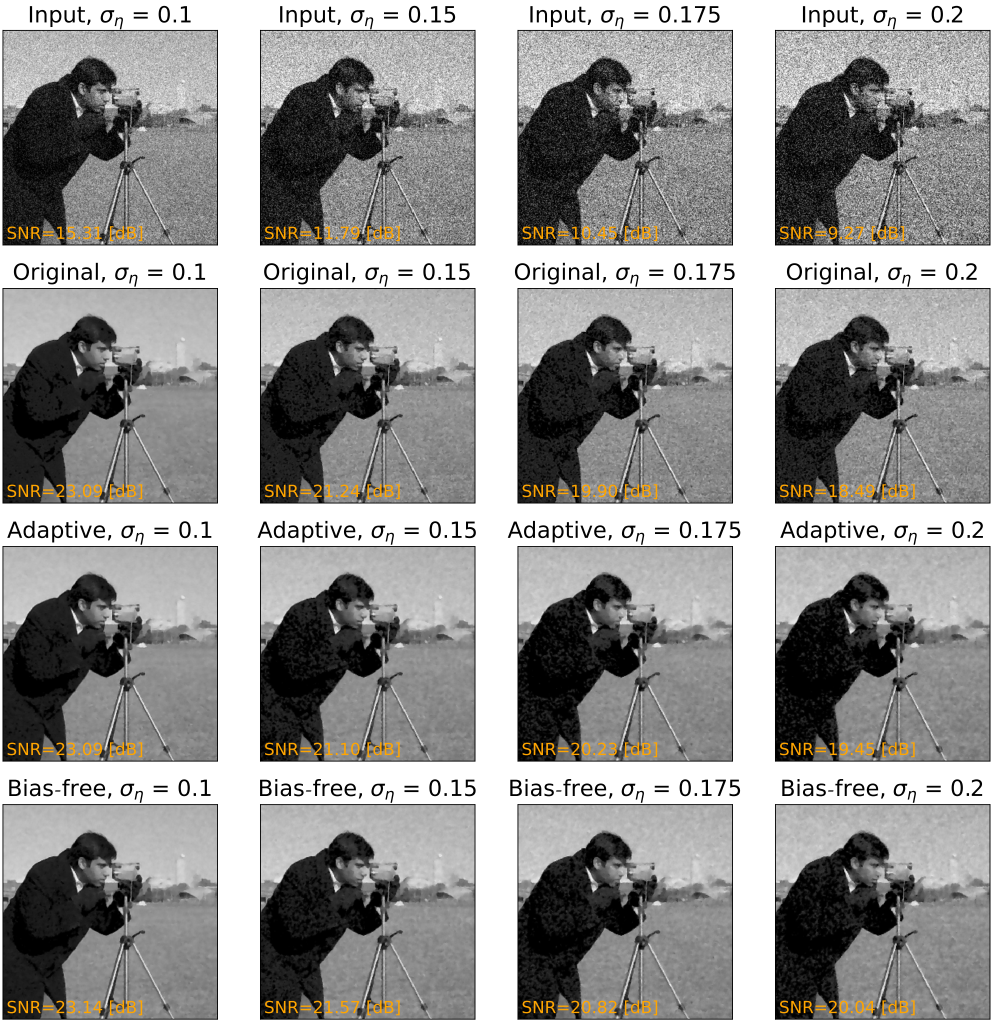

Input, σ_η = 0.1
Input, σ_η = 0.15
Input, σ_η = 0.175
Input, σ_η = 0.2
SNR=15.31 [dB]
SNR=11.79 [dB]
SNR=10.45 [dB]
SNR=9.27 [dB]
Original, σ_η = 0.1
Original, σ_η = 0.15
Original, σ_η = 0.175
Original, σ_η = 0.2
SNR=23.09 [dB]
SNR=21.24 [dB]
SNR=19.90 [dB]
SNR=18.49 [dB]
Adaptive, σ_η = 0.1
Adaptive, σ_η = 0.15
Adaptive, σ_η = 0.175
Adaptive, σ_η = 0.2
SNR=23.09 [dB]
SNR=21.10 [dB]
SNR=20.23 [dB]
SNR=19.45 [dB]
Bias-free, σ_η = 0.1
Bias-free, σ_η = 0.15
Bias-free, σ_η = 0.175
Bias-free, σ_η = 0.2
SNR=23.14 [dB]
SNR=21.57 [dB]
SNR=20.82 [dB]
SNR=20.04 [dB]

Figure 3.13. Comparison of the baseline (original) toy model against its adaptive and bias-free variants. The models are evaluated in the cameraman picture with increasing noise levels. The top row displays the noisy input. The second top-row represents the images processed with the original toy model. Meanwhile, the third row are the results of the adaptive toy model. Finally, the bottom-row are the results corresponding to the bias-free model. It can be observed that the performance original toy model degrades as the noise level increases, while the performance adaptive and bias-free model degrade less with increased noise levels, resulting in pictures with lower noise levels.

a complementary-phase tight framelet, then the pair $(\underline{\mathbf{K}}_l, \tilde{\underline{\mathbf{K}}}_l)$ *approximately* complies with the condition presented in Eq. (3.34), which for identify input $\mathbf{I}$ simplifies to

$$(\underline{\mathbf{K}}_n)_+ \circledast \tilde{\underline{\mathbf{K}}}_n = \mathbf{I} \cdot c_n \, , \tag{3.111}$$

in which c_n is an arbitrary constant.

The toy model is trained on images that contain randomly-generated overlapping triangles. All the images are scaled to the range [0,1]. For this experiment, the input to the images is the noise-contaminated image and the objective/desired output is the noiseless image. For training the CNNs, normally-distributed noise with a standard deviation of 0.1 is added to the ground truth. For every epoch, a batch of 192 training images is generated. For validation and test images, we use the "astronaut" and "cameraman" images included in the software package *Scipy*. The model is optimized with Adam for 25 epochs with a linearly decreasing learning rate. The initial learning rate for the optimizer is set to 10^{-3} and the batch size is set to 1 sample. The convolution kernels are initialized with Xavier initialization using a uniform distribution (see Glorot and Bengio [117]). The code is available at IEEE's code ocean `https://codeocean.com/capsule/7845737/tree`.

With the described settings, we have trained the toy model, and have tested if the phase-complementary tight-framelet property holds for the filters of the deepest level l=2. The results for the operation $(\underline{\mathbf{K}}_2)_+ \circledast \tilde{\underline{\mathbf{K}}}_2^\intercal$ are displayed in Fig. 3.10 (left), which shows that when the weights of the encoder and decoder have different initial values, the kernel pair ($\underline{\mathbf{K}}_2$, $\tilde{\underline{\mathbf{K}}}_2$) are not complementary-phase tight framelets. We have observed that the forward and inverse filters of wavelets/framelets are often the same or at least very similar. Based on this findings, we have initialized and trained the toy model again, now with the same initial values, for the kernel pairs ($\underline{\mathbf{K}}_n$, $\tilde{\underline{\mathbf{K}}}_n$). As shown by Fig. 3.10 (right), with the proposed initialization and subsequent training, the filters of the CNN converge to tensors with properties reminiscent of complementary-phase tight-framelets. This suggests that the initialization of the CNN has an important influence on the convergence of the model to a specific solution.

Fig. 3.12 displays a test image processed with two toy models, one trained trained with different and one trained with the same initial values for the encoding-decoding pairs. Note that the basic training consists in mapping a noisy input to an estimate of the noiseless signal. It can be observed that there are no significant differences between the images produced by both models. In the same figure (lower row), we have set the bias of both networks to zero. In this case, the network should reconstruct the noisy input, as confirmed by the figure, where both CNNs partly reconstruct the original noisy signal. This result suggests that the ReLU plus bias pairs operate akin to the low-rank approximation mechanism proposed the theory of deep convolutional framelets.

The following conclusions can be drawn from this experiment. First, the filters of the CNN may not necessarily converge to complementary-phase tight framelets. This is possibly caused by the initialization of the network and/or the interaction/co-adaptation between the multiple encoder/decoder layers. Second, we confirm that for our experimental setting, the low-rank approximation behavior in the CNN can be observed. For example, when setting the biases and thresholds to zero, part of the noise texture (high-rank information) is recovered, albeit the structure of the recovered noise did change with the initialization setting. Still, we are not sure if this matter could be resolved by more extensive training. Third, it is possible that linear filtering happens in the network as well, which may explain why noise texture is not fully recovered when setting the biases to zero. Fourth and final, we have observed that the behavior of the trained models change drastically depending on factors such as the learning rate and the initialization values of the model. For this reason, we consider this experiment and its outcome more as a proof of concept, where further investigation is needed.

3.6.2 Generalization

From the explanations in Section 3.3.3, it can be noted that the bias/threshold used in CNNs can control how much of the signal is suppressed by the nonlinearities. In addition, Section 3.4.4 established that there are additional mechanisms for noise reduction within the CNN, such as the Wiener-like behavior observed by Mohan *et al.* [104]. This raises the question how robust conventional CNNs are to noise-level changes different from the level that the model has been trained with. To address this question, we have trained two variants of the toy model. The first variant is inspired by the multi-scale sparse coding network by Mentl *et al.* [48], where the biases of each of the nonlinearities (ReLU in this case) are multiplied by an estimate of the standard deviation of the noise. In the design of this example, the noise estimate $\hat{\sigma}_\eta$, is defined in accordance with Chang *et al.* [14] and specified by

$$\hat{\sigma}_\eta = 1.4826 \cdot \text{Median}(|\mathbf{f}_{\text{HH}} * \mathbf{x}|). \tag{3.112}$$

Here, variable $\mathbf{f}_{\text{HH}}$ is the diagonal convolution filter of the discrete wavelet transform with Haar basis. For comparison purposes, we will refer to this model as the *adaptive* toy model. The second variant of the toy model being tested, examines the case where the convolution layers of the model do not add bias to the signal. This model is based in the so-called bias-free CNNs proposed by Mohan *et al.*, in which the bias of every convolution filter is set to zero during training. This setting has the purpose of achieving better generalization on the model, since it is claimed that this modification causes the model to behave independently of the noise level. This model will be referred to as the *bias-free* toy model.

We have trained the described variants of the toy models with the same settings of the experiment in Section 3.6.1. The three models are evaluated on the test image with varying noise levels $\sigma_n \in$[0.100, 0.150, 0.175, 0.200] and the result for this evaluation is displayed in Fig. 3.13. The result confirms that the performance of the original toy model degrades for higher noise levels. In contrast, the adaptive and bias-free models toy model perform better than the original toy model for most noise levels .

The results of this simple experiment confirm the diverse noise reduction mechanisms within a CNN, as well as showing that the CNNs have certain modeling limitations. For example, noise invariance, which can be addressed by further incorporating prior knowledge to the model, such as the case of the adaptive model, or by forcing the model to have a more Wiener-like behavior such as in the bias-free model. For the latter model, it should be noted that it is possible to obtain exactly the same behavior with the original toy model if the biases of the model would have converged to zero. This reasoning suggests that the large amount of free parameters and non-linear

Design elements		Express.	Perform.	No. param.	Rec. field
Activation	ReLU	High	Best	High	N/A
	Shrinkage	Low	Good	Medium	N/A
	Clipping	Low	Good	Medium	N/A
Scale	Single-scale	High	Good	High	Big
	Multi-scale	High	Good	Med./high	Small
Topology	Non-residual	High	Good	Higher	N/A
	Residual	High	Best	Lower	N/A

Table 3.2. Design elements and their impact in performance and computation cost.

behavior of the model can potentially prevent to find the optimal/robust solution, in which case the incorporation of prior knowledge can help to improve the model.

3.7 Which network fits to my problem?

3.7.1 Design elements

When choosing or designing a CNN for a specific noise reduction application, multiple choices and design elements should be considered. For example, the target performance, the memory required to train/deploy models, if certain signal preservation characteristics are required, the target execution time for the model, the characteristics of the images being processed, etc. Based on these requirements, diverse design elements of CNNs can be more or less desirable, for example, the activation functions, the use of single/multi-resolution models, the need for skip connections, and so forth. This section briefly discusses such elements by focusing on the impact that these elements have in terms of performance and potential computational cost. A summary of the main conclusions of these elements is included in Table 3.2.

A Nonlinearity

In literature, the most common activation function in CNNs is the ReLU. There are two main advantages of the ReLU with respect to other activations. First, ReLUs potentially enforce more sparsity in the feature maps than –for example– the soft shrinkage, because ReLUs cancel not only small values of the feature maps like the shrinkage functions do, but also all the negative values. The second advantage of the ReLU is its capacity to approximate other functions (see Section 3.4.3). Note that the high capacity of the ReLU to represent other functions [43], [118] (often referred to as *expressivity*) may also be one of the reasons why these models are prone to overfitting.

The better expressivity of the ReLU-CNNs may be the reason why –at the time of writing this manuscript– ReLU-based CNNs perform marginally better than the shrinkage-based models in terms of metrics such as signal-to-noise ratio or the structural similarity index metric (SSIM) [53], [58], [119]. Despite this small benefit, the visual characteristics of estimates produced by ReLU and shrinkage-based networks are very similar. Furthermore, the computational cost of ReLU-based designs is potentially higher than the costs of networks using shrinkage functions, because ReLU-based models require more feature maps to preserve the signal integrity. For example, the learned wavelet frame shrinkage network (LWFSN) that will be introduced in Chapter 5 achieves a performance very close to the FBPConvNet and the tight-frame U-Net for noise reduction in computed tomography, but only with a small fraction of the total trainable parameters, which allows for a faster and less computation-expensive model [58].

B Single/multi-scale designs

Single-scale models have the advantage that they avoid aliasing because no down/up-sampling layers are used, which comes at the expense of more computations and memory. Furthermore, this approach may lead to models with larger filters and/or deeper networks to achieve the same receptive field than multi-scale models do, which may further increase the computation costs of single-scale models. For example, in Table 3.1 it can be observed that a single-scale CNN such as RED CNN executes more computations than a simple multi-scale model such as the multi-scale sparse coding network.

In the case of multi-scale models, the main consideration should be that the down/up-sampling structure should allow perfect signal reconstruction to avoid introducing aliasing and/or distortion to the image estimates (e.g. the discrete wavelet transform in the tight-frame U-Net and in the multi-scale sparse coding network).

C (Non-)residual models

Residual noise reduction CNNs often perform better than their non-residual counterparts (e.g. the U-Net vs FBPConvNet and the LWFSN vs the rLWFSN). The reason may be that the trained models have more freedom to learn the filters, because the design does not need to learn to reconstruct the signal, but only to estimate the noise signal [40]. Also, it can be observed that non-residual models potentially need more parameters than residual networks, since the propagation/reconstruction of the noiseless signal is also dependent on the number of channels of the network.

3.7.2 State-of-the art of image denoising CNNs

Defining the state-of-the art in image denoising with CNNs is challenging for diverse reasons. First, there is a wide variety of available CNNs, which often are not compared to each other. Second, the suitability of a CNN for a given task may depend on the image and noise characteristics, such as noise distribution and (non-)stationarity. Third, the large amount of variables, in terms of e.g. optimization, data and data augmentation, adds reproducibility issues which further complicate making a fair comparison between all available models [17].

Despite of the mentioned challenges, there is some consensus in suggesting that DnCNN [9] or one of its variants can be regarded as the state of the art for image denoising. This network is residual, single-scale, while also using ReLU activations. Furthermore, the bias-free version of DnCNN [104] further extends the generalization of DnCNN by making it invariant to the noise level. Another state-of-the-art model is the multi-level wavelet CNN [120], which has a design very similar to the tight-frame U-Net [36]. Both of these models are based on the original U-Net design [8], but are deployed in residual configuration and the down/up-sampling structure is based on the discrete wavelet transform. In addition, the use of encoding-decoding CNNs used as prior/regularizer within model-based methods [121] have further improved the denoising power of single encoding-decoding CNNs. An example of this approach is the plug-and-play framework [122], where CNNs such as DnCNN are used as priors/proximal operators in image restoration tasks.

3.8 Conclusions

This chapter has analyzed the encoding-decoding structure of CNNs with basic signal processing principles, such as low-rank approximation and framelets. In order to achieve this, we have presented a set of basic signal principles and their mathematical formulations and have fitted them within the mathematical description of CNN models.

Meaning of the followed approach. By taking the adopted approach, we have able to improve modeling and the related analysis of CNN models in the following way.

- We have summarized Ye's [40] observation on the need for channels with inverted phase in ReLU CNNs in the so-called *phase-complementary tight framelet condition*, which has been mathematically summarized. This view is the key for improving the analysis of ReLU-based models because in previous work the effect of the non-linearity is often neglected (models are often assumed linear during the signal reconstruction analysis).
- The theory of deep convolutional framelets [40] as well as other related works [39], [40], [48] have been incorporated to provide a single, more general framework to analyze and design CNNs, which unifies shrinkage, clipping and ReLU-based designs. Hence, the model designer has more freedom to adopt one of these functions based on the requirements of the model to improve the modeling process.

Summarizing and from the principal point of view, the analysis is as follows. First, the encoder maps the incoming signal to a sparse representation, where a MAP estimate of the noiseless band is performed with an activation function. Afterwards, the decoder maps the estimated noiseless signals back to the original domain. The processing chain of encoding-decoding CNNs is akin to low-rank approximation and framelet shrinkage. In addition, it can be concluded also that the activations implicitly assume noise and signal characteristics of the feature maps.

Contributions. These new modeling and mathematical formulations have revealed that the following aspects of encoding-decoding CNNs. (1) Multiple signal processing concepts converge in the mathematical formulation encoding-decoding CNNs models. For example, the convolution and down/up-sampling structure of the encoder-decoder structure is akin to the framelet decomposition, the activation functions are rooted on conventional signal estimators. In addition, linear filtering also may happen within the model. (2) The activations implicitly assume noise and signal characteristics of the feature maps.

In summary, this chapter has presented the following contributions

1. *Unified mathematical notation*. We provide a unified and explicit notation which has direct correspondence with the schematic representations of CNNs. This notation involves up/down-sampling, explicit tensor convolutions for the filtering applied during framelet decompositions, etc. In addition, the used schematic representations are more akin to the schemes used to represent signal processing systems such as filter banks, which is more accurate than the conventional block-based representations of CNNs.
2. *Better-informed design choices*. An additional contribution of this chapter is the set heuristics and hints that can be leveraged by CNNs designers from the results in Section 3.7).

The results of this chapter are further explored in succeeding work. Chapters 4 explores the design of a noise reduction CNN, where the encoding and decoding paths are tight wavelet framelets and the trainable part of the network removes the noise from the sparse encoded signal. In contrast, Chapter 5 presents a design where the encoding and decoding paths are learned, but where it is possible to prove practically and theoretically that the encoding and decoding paths of the CNN allow for perfect signal reconstruction and where the non-linearities suppress the noise from the signal. Both of these designs are based on the heuristics defined in this chapter and on existing noise reduction CNNs.

CHAPTER 4

Data-driven denoising using non-trainable framelets

4.1 Introduction

Chapter 3 has explored the mathematical formulation of encoding-decoding CNNs from a low-rank approximation perspective. As starting point, this analysis uses the theory of deep convolutional framelets [40] and shows that the operation of CNNs is akin to conventional framelet-based denoising pipelines. More specifically, the convolution layers within the encoder and decoder structure provide a linear decomposition reminiscent of the forward and inverse framelet transforms. Furthermore, the biases and activations within the CNN eliminate the noisy segments of the signal in a process akin to soft-shrinkage and low-rank approximation.

When considering the signal processing interpretation of noise reduction CNNs, it can be observed that in order to achieve noise reduction, CNNs learn to decompose, reconstruct and denoise the signal, which is significantly more complex than conventional wavelet-based noise reduction methods, where non-trainable framelets map the input to a sparse domain, in which a shrinkage function is applied and afterwards, the inverse transform is applied to produce the final noiseless estimate.

Challenges. Rather than exploiting conventional CNNs, such as the residual encoder-decoder (RED) CNN [37] and the filtered back-projection network [50] analyzed in Chapter 3, this chapter explores the design of a CNN where the encoding-decoding path is composed by non-trainable framelets. This design choice leverages the sparse transformation and perfect reconstruction provided by existing transformations and limits the operations that are learned to eliminate the noise in the signal. This approach poses several challenges listed below.

1. *Choice of framelet basis*. Given that this chapter explores a design that uses non-trainable filters for the encoder and decoder, the choice of appropriate basis functions is critical for the design to achieve a good performing model.
   (a) *Design towards directionality and sparsity*. Conventional orthogonal transforms are prone to aliasing and have limited directionality, which constrains their performance for noise reduction applications. In our application, we aim for improved directionality in order to preserve the image contents.
   (b) *Computational complexity*. The chosen framelet bases should have short

filters to improve the spatial localization of the design and to decrease the computational cost of training/executing the design. It should be noted that this comes at the expense of decreasing the frequency localization, hence there is a careful trade-off to be made in order to find the right basis[1]

2. *Design of the noise reduction structure.* As discussed in Section 3.3.3, there are multiple activation functions which can be applied for noise reduction applications. Since framelets provide a sparse representation, the activation functions often shrink low-amplitude components which are assumed to be noisy. Consequently, the choice of the appropriate function is critical for the optimal performance of the proposed model. In this chapter, rather than a single activation layer, we explore to incorporate a CNN to eliminate the noise of the sub-bands of the signal in the transformed domain.

The following section addresses the background on signal transformations in more depth and their associated noise reduction techniques. In that discussion, the above terms on directionality, sparsity and shrinkage techniques are frequently used to explore and assess the relation to noise reduction. The layout of the chapter will be presented at the end of Section 4.2.

4.2 Background on transformations and related work

In order to address noise reduction with framelets, multiple approaches have been proposed. Among the simplest pipelines, we find conventional models that use (bi)orthogonal transforms, in which the signal is converted to the framelet domain with a multi/single-resolution transform, which is also *sparse*. In the sparse domain, an element-wise sparsity enforcing function such as soft/hard thresholding is applied. These functions suppress samples with low amplitude, which are typically associated to noise. After the shrinkage stage, the signal is mapped back to the original domain to produce the final noiseless estimate. An example of a model that operates this way is the well-known Bayes-shrink [14]. More sophisticated denoising algorithms go beyond using the amplitude of the components to eliminate noise. An example of a more sophisticated system is Bi-shrink [76]. This system leverages the correlation between the detail information at multiple scales in multi-resolution framelet decompositions. Alternative approaches leverage information at the same scale. This is covered by for example, the locally-spatial indicators introduced in [123] and the well-known Markov random fields [124]. In addition to conventional wavelet-based denoising, also algorithms based on optimization techniques have been employed. Examples of these techniques are the (fast) iterative soft thresholding algorithm [109], [125].

A common limitation of the previously mentioned approaches can be attributed to the choice of (bi)orthogonal transforms, which are critically-sampled. This means that these transforms are implemented with filter banks with only enough bandwidth to

[1] When using signal transformations, the designer is always faced with the uncertainty principle. For example, in the wavelet domain, increasing the frequency localization of the wavelet bands comes at the expense of using longer filters that have lower spatial localization.

represent the original signal. Furthermore, as discussed in Fig. 3.3 of Section 3.3.2-B, the structure of (bi)orthogonal framelet transforms introduce aliasing in the transformed signal, which is cancelled in the inverse-transformed signal by carefully designing the filters of the forward and inverse transform. Consequently, when eliminating samples in the transformed domain as it is performed by conventional wavelet-based denoising algorithms, the relationship between the channels which allows for alias cancellation is broken and aliasing is introduced to the signal. In addition to aliasing, an additional limitation of (bi)orthogonal framelet transformations is the lack of directionality [126]. It should be noted that the filters of the conventional discrete wavelet transform are aligned only to 90°, 0° and ±45°, so that they have limited directionality. This causes that the sparsity in the transformed domain decreases for edges/transitions that have different directions than the previously mentioned angles, thereby limiting the performance of the noise reduction algorithms that use (bi)orthogonal transforms.

The limitations of (bi)orthogonal transforms have been solved by using *directional* transforms that are *redundant*. In contrast with critically-sampled transforms such as the *discrete wavelet transform* (DWT), redundant transforms are *oversampled*. This means that each of the bands in the transform represents a smaller region of the Fourier spectrum. Consequently, less aliasing is introduced by the transform [126], since there is more bandwidth available to represent the signal. In addition, the directionality of some of these transforms improves the performance of noise reduction algorithms. Examples of these transforms are curvelets [127], shearlets [128], the quaternion wavelet transform [129], the hyper-analytic wavelet transform [130], the directional hyper-complex wavelet transform [131] and, finally, the dual-tree complex wavelet transform [74], all of which have been applied to noise reduction applications.

This chapter particularly concentrates on the dual-tree complex wavelet transform (DTCWT). This transformation is an extension of the orthogonal discrete wavelet transform in which an additional wavelet tree is used for every dimension (often referred to as *trees* “a" and “b"). The additional tree has a group delay of half a sample with respect to the first tree, which causes the filtered signals to form an approximately analytic signal. In order to achieve the required phase shift, a common implementation is the so-called q-shift filters, which are orthogonal filters having a group delay of one quarter sample. The filter in tree “a" is inverted with respect to tree “b", which generates the required phase delay of a half sample.

The DTCWT has two attractive properties of the DTCWT, which are specified as follows. (1) The representation is an *analytic signal* which is is free of aliasing. (2) The implementation of the DTCWT is achieved with linear combinations of orthogonal DWT trees. The advantages of the DTCWT come with the backside aspects such as the following. (3) In order to achieve an analytic representation, the transform requires larger filters, which decreases the spatial localization property of the transform. Consequently, shrinking the wavelet components in this representation impacts a larger neighborhood in the image, which may be undesirable for noise-reduction applications. (4) The implementation of this transform is sequential, which may increase the execution time of the transformation.

Followed approach in this chapter. The design part of this chapter consists of two parts. (1)

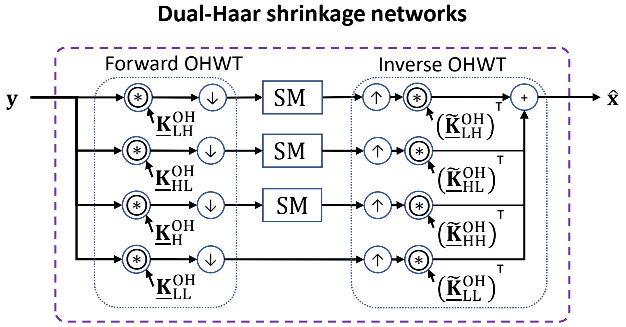

Dual-Haar shrinkage networks
Forward OHWT
Inverse OHWT
y
SM
SM
SM
$\hat{\mathbf{x}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LH}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HL}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{H}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LL}}$
$(\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LH}})^{\mathrm{T}}$
$(\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HL}})^{\mathrm{T}}$
$(\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HH}})^{\mathrm{T}}$
$(\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LL}})^{\mathrm{T}}$

Figure 4.1. High-level representation of a single-decomposition level DHSN model. In the figure, the SM block stands for shrinkage module, while the filters $\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LL}}$, $\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LH}}$, $\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HL}}$ and $\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HH}}$ are the filters of the overcomplete Haar wavelet transform for the low-pass and high-pass filters in the horizontal, vertical and diagonal directions, respectively. Furthermore, filters $\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LL}}$, $\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LH}}$, $\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HL}}$ and $\tilde{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HH}}$ are the corresponding filters for the inverse transform/decoder.

A fixed framelet basis known as the *overcomplete Haar wavelet transform* (OHWT) which transforms the signal to a sparse decomposition. (2) A noise-reduction CNN which eliminates the noise in the sparse domain. The combination of these two elements is referred to here as *dual-Haar shrinkage network* (DHSN), of which a high-level schematic description is depicted in Fig. 4.1.

The remainder of this chapter is structured as follows. Section 4.3 introduces the dual-tree complex wavelet transform and its modification to the wavelet transformation used in this chapter. Furthermore, it also introduces the design of two noise reduction modules. Section 4.4 shows the experiments and results of the proposed networks and compare them with RED CNN and FBPConvNet. Finally, Section 4.5 summarizes the main contributions of our work and addresses future endeavors.

4.3 Methods

The design developed in this section integrates a fixed framelet with noise reduction CNNs that eliminate the noise in the sparse domain. In order to achieve this endeavor, Sections 4.3.1 and 4.3.2 first explain the design of the framelet transform, while Section 4.3.3 addresses the design of the noise reduction CNNs.

The layout for this section is as follows. First, Section 4.3.1 addresses in more detail the design of the DTCWT. Afterwards, in Section 4.3.2, the algorithm of DTCWT is modified to define the overcomplete Haar wavelet transform. Finally, having defined the transformation used for our CNN, Section 4.3.3 addresses the design of the shrinkage networks employed in the designs here presented.

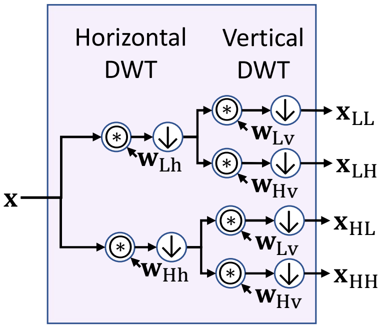

Horizontal DWT
Vertical DWT
$\mathbf{x}$
$\mathbf{w}_{\mathrm{Lh}}$
$\mathbf{w}_{\mathrm{Hh}}$
$\mathbf{w}_{\mathrm{Lv}}$
$\mathbf{w}_{\mathrm{Hv}}$
$\mathbf{w}_{\mathrm{Lv}}$
$\mathbf{w}_{\mathrm{Hv}}$
$\mathbf{x}_{\mathrm{LL}}$
$\mathbf{x}_{\mathrm{LH}}$
$\mathbf{x}_{\mathrm{HL}}$
$\mathbf{x}_{\mathrm{HH}}$

Figure 4.2. Conventional discrete wavelet transform tree for two dimensions. Filters $\mathbf{w}^{\mathrm{a}}_{\mathrm{Hh}}$ and $\mathbf{w}^{\mathrm{a}}_{\mathrm{Hv}}$ are the high-pass filters of the 1D DWT in the vertical and horizontal directions, respectively. Furthermore, the down-sampling represented by the arrow symbol $\downarrow$ is applied either to the horizontal or vertical dimensions.

4.3.1 Dual-tree complex wavelet (framelet) transform

Section 3.3.2-B shows that orthogonal framelets, such as the discrete wavelet transform (DWT), provide a sparse representation which is often used in noise reduction applications. For N-dimensional DWTs, the transformations (bi)orthonormal bases can be achieved by successive 1D transformations applied successively along each each of the N dimensions. The resulting filtering structure for every dimension is referred to as a *tree*.

As mentioned in Section 4.1, the conventional DWT is in limited directionality and introduces aliasing, which limits its performance for image denoising. These shortcomings are circumvented by redundant directional transforms such as the DTCWT [74]. This transform introduces an additional DWT tree per dimension (referred to as tree “a" and “b"), where the filters of the tree “b" have a group delay of half a sample with respect to the tree “a". This means that the tree “b" approximates a quadrature component of tree “a". Consequently, if tree “b" is considered to be a complex/imaginary number, then the sum of trees “a" and “b" form an approximately analytic signal (i.e. it has a Fourier spectrum which is only positive). In addition, by grouping the real and complex components of the transform, a set of directional filters are generated, which improves the representation of image segments that are not aligned in the horizontal or vertical axes. In summary, the algorithm to compute the DTCWT, is composed by the following steps.

1. The input signal $\mathbf{x}$ is processed with the the trees “a" and “b" for every dimension and for every wavelet band, except for the low-frequency (LL). In 2D, this process generates a four-channel system per wavelet band. In this configuration, for every tree “b", it is assumed that the signal is multiplied by a complex number $\gamma = \pm j|\gamma|$, where j^2=-1.

   It should be noted that Kingsbury [74] only applies the transform with q-shift filters for decomposition levels higher than one. In the first decomposition level, any orthogonal or (bi)orthogonal wavelet can be used, as long as both trees are shifted by one sample with respect to each other, which virtually generates an

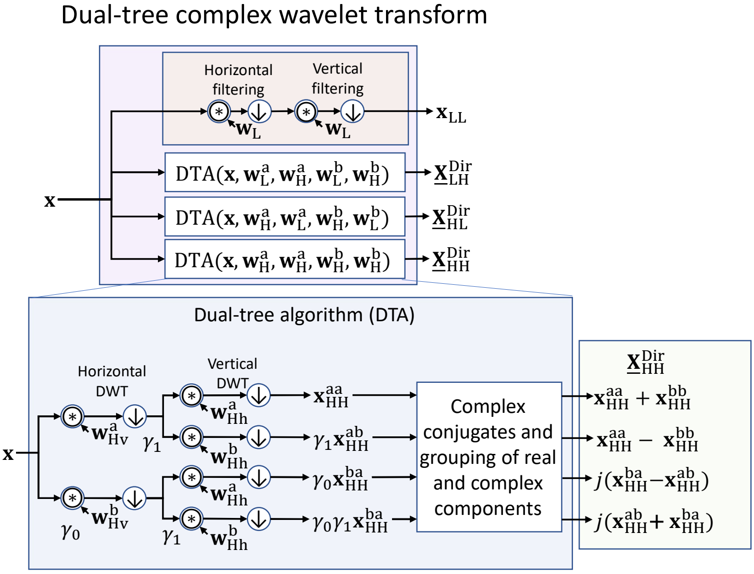

Dual-tree complex wavelet transform
Horizontal filtering
Vertical filtering
$\mathbf{w}_{\mathrm{L}}$
$\mathbf{w}_{\mathrm{L}}$
$\mathbf{x}_{\mathrm{LL}}$
$\mathbf{x}$
$\mathrm{DTA}(\mathbf{x}, \mathbf{w}^{\mathrm{a}}_{\mathrm{L}}, \mathbf{w}^{\mathrm{a}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{L}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{H}})$
$\underline{\mathbf{X}}^{\mathrm{Dir}}_{\mathrm{LH}}$
$\mathrm{DTA}(\mathbf{x}, \mathbf{w}^{\mathrm{a}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{a}}_{\mathrm{L}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{L}})$
$\underline{\mathbf{X}}^{\mathrm{Dir}}_{\mathrm{HL}}$
$\mathrm{DTA}(\mathbf{x}, \mathbf{w}^{\mathrm{a}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{a}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{H}}, \mathbf{w}^{\mathrm{b}}_{\mathrm{H}})$
$\underline{\mathbf{X}}^{\mathrm{Dir}}_{\mathrm{HH}}$
Dual-tree algorithm (DTA)
Horizontal DWT
Vertical DWT
$\mathbf{x}$
$\mathbf{w}^{\mathrm{a}}_{\mathrm{Hv}}$
γ_1
$\mathbf{w}^{\mathrm{b}}_{\mathrm{Hv}}$
γ_0
γ_1
$\mathbf{w}^{\mathrm{a}}_{\mathrm{Hh}}$
$\mathbf{w}^{\mathrm{b}}_{\mathrm{Hh}}$
$\mathbf{w}^{\mathrm{a}}_{\mathrm{Hh}}$
$\mathbf{w}^{\mathrm{b}}_{\mathrm{Hh}}$
$\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}}$
$\gamma_1\mathbf{x}^{\mathrm{ab}}_{\mathrm{HH}}$
$\gamma_0\mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}}$
$\gamma_0\gamma_1\mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}}$
Complex conjugates and grouping of real and complex components
$\underline{\mathbf{X}}^{\mathrm{Dir}}_{\mathrm{HH}}$
$\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}} + \mathbf{x}^{\mathrm{bb}}_{\mathrm{HH}}$
$\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}} - \mathbf{x}^{\mathrm{bb}}_{\mathrm{HH}}$
$j(\mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}} - \mathbf{x}^{\mathrm{ab}}_{\mathrm{HH}})$
$j(\mathbf{x}^{\mathrm{ab}}_{\mathrm{HH}} + \mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}})$

Figure 4.3. Graphical representation of the algorithm used to represent the dual-tree complex wavelet transform.

undecimated transform.

2. The bands generated by the dual-tree configuration are added and grouped in real and complex components. Furthermore, each individual real or complex component is considered to be an independent signal and placed on a separate image channel. This process is repeated for the complex conjugates of tree “b".

To exemplify the algorithm of the dual-tree complex wavelet transform we refer to Fig. 4.2, where the conventional DWT is described. In contrast, Fig. 4.3 showcases the DTCWT, where it can be observed that the dual-tree algorithm is applied to the LH, HL and HH bands. Furthermore, for better understanding it is shown that the processing of the signal x with the tree “a" in the horizontal dimension and “a" for the band HH results in the signal $\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}}$, defined by

$$\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}} = f_{(2\downarrow\mathrm{v})}(f_{(2\downarrow\mathrm{h})}(\mathbf{x} * \mathbf{w}^{\mathrm{a}}_{\mathrm{Hh}}) * \mathbf{w}^{\mathrm{a}}_{\mathrm{Hv}}). \tag{4.1}$$

Here, $f_{(2\downarrow\mathrm{v})}(\cdot)$ and $f_{(2\downarrow\mathrm{h})}(\cdot)$ are the down-sampling functions in the vertical and horizontal directions, respectively, while $\mathbf{w}^{\mathrm{a}}_{\mathrm{Hh}}$ and $\mathbf{w}^{\mathrm{a}}_{\mathrm{Hv}}$ are the high-pass filters of the 1D DWT in the vertical and horizontal directions. When processing all the branches of the dual-tree and grouping the components in real and complex values leads to

$$(\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}} + \gamma_0\gamma_1\mathbf{x}^{\mathrm{bb}}_{\mathrm{HH}}) + (\gamma_1\mathbf{x}^{\mathrm{ab}}_{\mathrm{HH}} + \gamma_0\mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}}), \tag{4.2}$$

where $\gamma_n = \pm j$. Assuming $\gamma_0 = j$ and $\gamma_1 = -j$ generates the signal

$$(\mathbf{x}^{\mathrm{aa}}_{\mathrm{HH}} + \mathbf{x}^{\mathrm{bb}}_{\mathrm{HH}}) + j \cdot (-\mathbf{x}^{\mathrm{ab}}_{\mathrm{HH}} + \mathbf{x}^{\mathrm{ba}}_{\mathrm{HH}}). \tag{4.3}$$

Furthermore, if $\gamma_0 = j$ and $\gamma_1 = +j$, the signal becomes

$$(\mathbf{x}_{\mathrm{HH}}^{\mathrm{aa}} - \mathbf{x}_{\mathrm{HH}}^{\mathrm{bb}}) + j \cdot (\mathbf{x}_{\mathrm{HH}}^{\mathrm{ab}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{ba}}). \tag{4.4}$$

From Eq. (4.3) and Eq. (4.4), we obtain two real terms $(\mathbf{x}_{\mathrm{HH}}^{\mathrm{aa}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{bb}})$ and $(\mathbf{x}_{\mathrm{HH}}^{\mathrm{aa}} - \mathbf{x}_{\mathrm{HH}}^{\mathrm{bb}})$ and two complex terms $(-\mathbf{x}_{\mathrm{HH}}^{\mathrm{ab}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{ba}})$ and $(\mathbf{x}_{\mathrm{HH}}^{\mathrm{ab}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{ba}})$. which can be grouped as a single tensor $\underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{Dir}}$, defined by

$$\underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{Dir}} = \left((\mathbf{x}_{\mathrm{HH}}^{\mathrm{aa}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{bb}}) \quad (\mathbf{x}_{\mathrm{HH}}^{\mathrm{aa}} - \mathbf{x}_{\mathrm{HH}}^{\mathrm{bb}}) \quad (-\mathbf{x}_{\mathrm{HH}}^{\mathrm{ab}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{ba}}) \quad (\mathbf{x}_{\mathrm{HH}}^{\mathrm{ab}} + \mathbf{x}_{\mathrm{HH}}^{\mathrm{ba}})\right). \tag{4.5}$$

Here, the complex number j has been excluded since the implementations are often based on real arithmetic (where j is inferred by the channel location). As final step, the full DTCWT transform of a signal is specified in tensor form as previously by

$$\mathrm{DTCWT}(\mathbf{x}) = \left(f_{(2\downarrow)}(\mathbf{x} \circledast \mathbf{w}_{\mathrm{LL}}) \quad \underline{\mathbf{X}}_{\mathrm{LH}}^{\mathrm{Dir}} \quad \underline{\mathbf{X}}_{\mathrm{HL}}^{\mathrm{Dir}} \quad \underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{Dir}}\right). \tag{4.6}$$

Here, $\underline{\mathbf{X}}_{\mathrm{LH}}^{\mathrm{Dir}}$ and $\underline{\mathbf{X}}_{\mathrm{HL}}^{\mathrm{Dir}}$ are computed with the same algorithm that is previously presented for $\underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{Dir}}$. Finally, as a closing remark, it should be noted that in order to recover the original signal, it is sufficient to apply all the operations in the reverse order.

4.3.2 Overcomplete Haar wavelet transform (OHWT)

A major limitation of the DTCWT is the longer support of the q-shift filters when compared with the DWT. It should be noticed that longer filters decrease the spatial localization of the transformation. Furthermore, an additional downside of the DTCWT is the sequential nature of the algorithm used to compute it, which makes it less interesting for its integration into CNNs, since it would increase the execution time of models based on this transform.

In order to reduce the kernel size of the filters used on the DTCWT, we have modified the original algorithm to the so-called *over-complete Haar wavelet transform* (OHWT). The OHWT uses the dual-tree structure of the first decomposition level of the DTCWT and applies it to *all* the bands instead of only doing it to the detail as performed in the DTCWT. This means that the filters in trees “a" and “b" are the same, but they are shifted by one sample in every dimension. Furthermore, the OHWT uses the Haar basis as convolution kernel which is very compact. The motivation for the proposed modifications are twofold. First, this transform preserves the directionality of the DTCWT, while offering a compact response (equivalent to convolution masks of 3×3 samples). Second, the transform is virtually undecimated, which means that less aliasing components will occur in the noise reduction application. Finally, as will be shown, this structure simpler to implement in a real practical system, because the full transform can be summarized in a single convolution basis, rather than the sequential algorithm of the DTCWT.

With the aforementioned modifications to the original DTCWT algorithm, the OHWT changes the DTCWT-converted signal from Eq. (4.5) to

$$\underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{OHWT}} = \begin{pmatrix} f_{(2\downarrow)}(\ \ \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{aa}} + \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{bb}}) \\ f_{(2\downarrow)}(\ \ \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{aa}} - \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{bb}}) \\ f_{(2\downarrow)}(-\mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{ab}} + \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{ba}}) \\ f_{(2\downarrow)}(\ \ \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{ab}} + \mathbf{x} * \mathbf{w}_{\mathrm{HH}} * \mathbf{s}_{\mathrm{ba}}) \end{pmatrix}^{\mathsf{T}}, \tag{4.7}$$

where the shifts are introduced by convolving with the convolution masks s_{aa}, s_{ab}, s_{ba} and s_{bb}. The filtering coefficients are found in the filter $\mathbf{w}_{\mathrm{HH}}$. Specifically, s_{aa} is the identity operation, s_{ab} provides a shift of one sample in the vertical dimension, s_{ba} provides a shift of one sample in the horizontal dimension, s_{ab} is a shift in the horizontal dimension and s_{bb} is a shift in both, the vertical and horizontal dimensions. The shift operators s_{aa}, s_{ab}, s_{ba} and s_{bb} are defined by

$$\mathrm{s}_{\mathrm{aa}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \mathrm{s}_{\mathrm{ab}} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad \mathrm{s}_{\mathrm{ba}} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \mathrm{s}_{\mathrm{bb}} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{4.8}$$

Consequently, Eq. (4.7) can be simplified in notation to

$$\underline{\mathbf{X}}_{\mathrm{HH}}^{\mathrm{OHWT}} = f_{(2\downarrow)}(\mathbf{x} \circledast \mathbf{w}_{\mathrm{HH}} \circledast \underline{\mathbf{S}}), \tag{4.9}$$

where the convolution kernel $\underline{\mathbf{S}}$ is given by

$$\underline{\mathbf{S}} = \begin{pmatrix} (\ \ \mathrm{s}_{\mathrm{aa}} + \mathrm{s}_{\mathrm{bb}}) \\ (\ \ \mathrm{s}_{\mathrm{aa}} - \mathrm{s}_{\mathrm{bb}}) \\ (-\mathrm{s}_{\mathrm{ab}} + \mathrm{s}_{\mathrm{ba}}) \\ (\ \ \mathrm{s}_{\mathrm{ab}} + \mathrm{s}_{\mathrm{ba}}) \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{pmatrix}^{\mathsf{T}}. \tag{4.10}$$

Here, when inspecting the result of Eq. (4.10), it becomes explicit that the combination of the dual-tree structure and the linear combinations of the OHWT generate two DWTs with 1D Haar bases that are rotated by ±45° (see the 2×2 matrices at the right, therefore we now use the symbol $\underline{\mathbf{H}}$). This result is very convenient, because it means that whole transform can be implemented with fixed convolutional layers, which are readily available in standard deep learning (DL) libraries. Applying the same procedure to all the wavelet bands, we achieve that the OHWT can be summarized to a single tensor convolution, or equivalently

$$\underline{\mathbf{x}}_{\mathrm{OHWT}} = f_{(2\downarrow)}(\mathbf{x} \circledast \underline{\mathbf{K}}_{\mathrm{OH}}), \tag{4.11}$$

in which convolution kernel $\underline{\mathbf{K}}_{\mathrm{OH}}$ is defined by

$$\underline{\mathbf{K}}_{\mathrm{OH}} = \begin{pmatrix} \underline{\mathbf{K}}_{\mathrm{LL}}^{\mathrm{OH}} \\ \underline{\mathbf{K}}_{\mathrm{LH}}^{\mathrm{OH}} \\ \underline{\mathbf{K}}_{\mathrm{HL}}^{\mathrm{OH}} \\ \underline{\mathbf{K}}_{\mathrm{HH}}^{\mathrm{OH}} \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} (\mathbf{w}_{\mathrm{LL}} \circledast \underline{\mathbf{H}}) \\ (\mathbf{w}_{\mathrm{LH}} \circledast \underline{\mathbf{H}}) \\ (\mathbf{w}_{\mathrm{HL}} \circledast \underline{\mathbf{H}}) \\ (\mathbf{w}_{\mathrm{HH}} \circledast \underline{\mathbf{H}}) \end{pmatrix}^{\mathsf{T}}. \tag{4.12}$$

Here, the specification of the numerical values of tensor $\underline{\underline{\mathbf{K}}}_{\mathrm{OH}}$ are given by

$$\left(\begin{matrix}
\left(\begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}\right) \\
\left(\begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ 1 & -2 & 1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1. \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ -1 & 2 & -1 \\ -1 & 1 & 0 \end{pmatrix}\right) \\
\left(\begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ -1 & -2 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 & -1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ -1 & -2 & -1 \\ 1 & 1 & 0 \end{pmatrix}\right) \\
\left(\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & -1 \\ 1 & -2 & 1 \\ -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}\right)
\end{matrix}\right) \frac{1}{2\sqrt{2}},$$

where the kernel $\underline{\underline{\mathbf{K}}}_{\mathrm{OH}}$ is composed by four sub-tensors $\underline{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LL}}$, $\underline{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LH}}$, $\underline{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HL}}$ and $\underline{\underline{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HH}}$ with dimensions $(1 \times 4 \times 3 \times 3)$. A graphical representation of OHWT is shown in Fig. 4.4.

In order to map the representation $\underline{\underline{\mathbf{x}}}_{\mathrm{OHWT}}$ back to its original signal $\mathbf{x}$, an analogous procedure to the forward transform is required, which results in an inverse OHWT convolution kernel $\tilde{\underline{\underline{\mathbf{K}}}}_{\mathrm{OH}}$. Just as the forward transformation kernel can be summarized in tensor $\tilde{\underline{\underline{\mathbf{K}}}}$, or in in term of its components $\tilde{\underline{\underline{\mathbf{K}}}}^{\mathrm{OH}}_{\mathrm{LL}}$, $\hat{\underline{\underline{\mathbf{K}}}}^{\mathrm{OH}}_{\mathrm{LH}}$, $\tilde{\underline{\underline{\mathbf{K}}}}^{\mathrm{OH}}_{\mathrm{HL}}$ and $\tilde{\underline{\underline{\mathbf{K}}}}^{\mathrm{OH}}_{\mathrm{HH}}$. The numerical values of $\tilde{\underline{\underline{\mathbf{K}}}}_{\mathbf{OH}}$ are defined by

$$\left(\left(\left(\begin{matrix}
\begin{pmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \\
\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} & \begin{pmatrix} 1 & -1 & 0 \\ 1 & -2 & 1 \\ 0 & -1 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 & -1 \\ -1 & 2 & -1 \\ -1 & 1 & 0 \end{pmatrix} \\
\begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix} & \begin{pmatrix} 1 & 1 & 0 \\ -1 & -2 & -1 \\ 0 & 1 & 1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 & 1 \\ -1 & -2 & -1 \\ 1 & 1 & 0 \end{pmatrix} \\
\begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix} & \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix} & \begin{pmatrix} 0 & 1 & -1 \\ 1 & -2 & 1 \\ -1 & 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}
\end{matrix}\right)\right)\right) \frac{1}{8\sqrt{2}}. \tag{4.13}$$

It should be noticed that the OHWT, is a tight framelet which allows for perfect reconstruction. Therefore, its inverse satisfies the formulation

$$\mathbf{x} = f_{(2\uparrow)}(\underline{\underline{\mathbf{x}}}_{\mathrm{OHWT}}) \circledast \tilde{\underline{\underline{\mathbf{K}}}}^{\mathsf{T}}_{\mathrm{OH}}. \tag{4.14}$$

A reflection on the achieved formulation of the forward/backward OHWT is helpful at this point. The OHWT is also conceptually similar to the directional hypercomplex (DHWT) and quaternion wavelet (QWT) transforms [132]. However, the OHWT is not an analytic representation, which is the case with the DHWT, QWT and

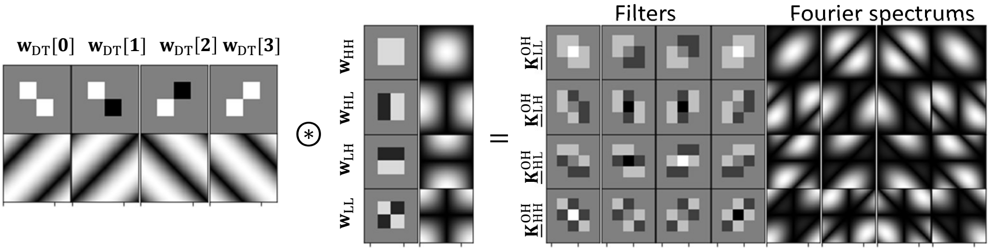

Filters
Fourier spectrums
$\mathbf{w}_{\mathrm{DT}}[\mathbf{0}]$
$\mathbf{w}_{\mathrm{DT}}[\mathbf{1}]$
$\mathbf{w}_{\mathrm{DT}}[\mathbf{2}]$
$\mathbf{w}_{\mathrm{DT}}[\mathbf{3}]$
$\circledast$
$\mathbf{w}_{\mathrm{HH}}$
$\mathbf{w}_{\mathrm{HL}}$
$\mathbf{w}_{\mathrm{LH}}$
$\mathbf{w}_{\mathrm{LL}}$
=
$\underline{\mathbf{K}}_{\mathrm{LL}}^{\mathrm{OH}}$
$\underline{\mathbf{K}}_{\mathrm{LH}}^{\mathrm{OH}}$
$\underline{\mathbf{K}}_{\mathrm{HL}}^{\mathrm{OH}}$
$\underline{\mathbf{K}}_{\mathrm{HH}}^{\mathrm{OH}}$

Figure 4.4. Visualization of the generation of the kernels of the OHWT. At the left, the diagonal Haar wavelets are shown, in the middle the Haar basis for the 2D DWT are displayed. Finally, the right subfigure shows the resulting 2D basis functions. Note that the kernel $\underline{\mathbf{K}}_{\mathrm{OH}}$ has good space and frequency localization, while the filters are also directional.

DTCWT. Furthermore, in the case of the 2D DTCWT, DHWT and the DTCWT, the additional transformations that are required to generate the directional filters are only applied to the high-frequency bands, while in the OHWT the same transformations are applied to all the bands, including the low-frequency one.

4.3.3 Wavelet Shrinkage Networks (WSNs)

The proposed wavelet shrinkage networks (WSNs) are generated by integrating the encoding path defined by Eq. (4.12), the decoding section is specified by Eq. (4.14), as well as the processing path executing wavelet shrinkage, denoted by CNN$(\cdot)$, operated on the detail bands LH, HL and HH. As a result, a one-decomposition-level WSN is expressed by

$$\begin{aligned} G(\mathbf{x}) =& f_{(2\uparrow)}(f_{(2\downarrow)}(\mathbf{x} \circledast \underline{\mathbf{K}}_{\mathrm{LL}}^{\mathrm{OH}})) \circledast (\tilde{\underline{\mathbf{K}}}_{\mathrm{LL}}^{\mathrm{OH}})^{\intercal}+ \\ & f_{(2\uparrow)}(\mathrm{SN}(f_{(2\downarrow)}(\mathbf{x} \circledast \underline{\mathbf{K}}_{\mathrm{LH}}^{\mathrm{OH}}))) \circledast (\tilde{\underline{\mathbf{K}}}_{\mathrm{LH}}^{\mathrm{OH}})^{\intercal}+ \\ & f_{(2\uparrow)}(\mathrm{SN}(f_{(2\downarrow)}(\mathbf{x} \circledast \underline{\mathbf{K}}_{\mathrm{HL}}^{\mathrm{OH}}))) \circledast (\tilde{\underline{\mathbf{K}}}_{\mathrm{HL}}^{\mathrm{OH}})^{\intercal}+ \\ & f_{(2\uparrow)}(\mathrm{SN}(f_{(2\downarrow)}(\mathbf{x} \circledast \underline{\mathbf{K}}_{\mathrm{HH}}^{\mathrm{OH}}))) \circledast (\tilde{\underline{\mathbf{K}}}_{\mathrm{HH}}^{\mathrm{OH}})^{\intercal}. \end{aligned} \tag{4.15}$$

In the first term of Eq. (4.15), the operator SN$(\cdot)$ is missing because shrinkage is not applied in that sub-band.

For deeper CNNs, the OHWT is applied recursively using the low-frequency band as input for the next wavelet decomposition. It is important to bear in mind that the OHWT is in fact an *expansive* representation. Therefore, for every input channel in the signal, four image channels are generated for each of the bands. This means that if deeper networks are required, an exponential growth in image channels/feature maps is faced, which is described by the expression

$$N(C, L) = C \cdot 4^{L}, \tag{4.16}$$

in which N represents the number of feature maps per wavelet band at the decomposition level L >1 given an input signal with C channels. It should be noticed that these are the values per band, and in 2D, the OHWT decomposes the signal into four bands.

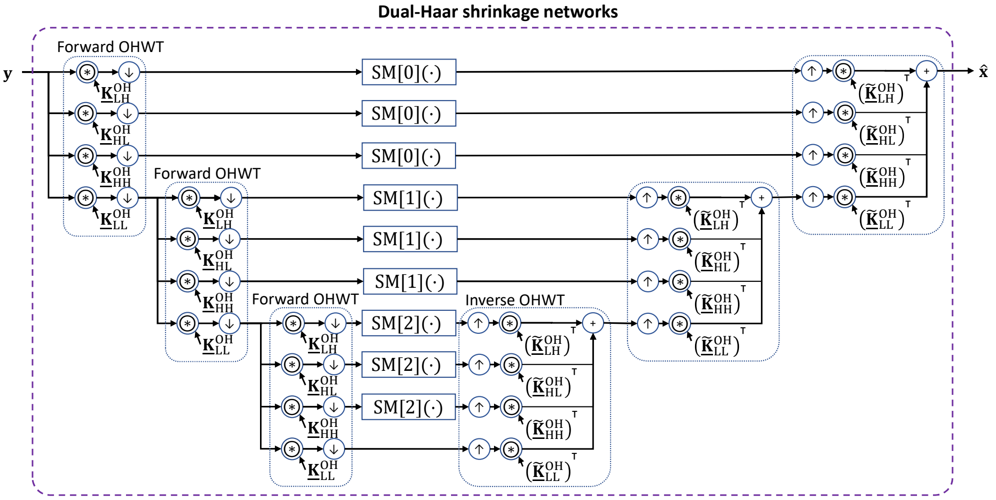

Dual-Haar shrinkage networks
Forward OHWT
y
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LH}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HL}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HH}}$
$\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LL}}$
Forward OHWT
Forward OHWT
Inverse OHWT
SM[0](·)
SM[0](·)
SM[0](·)
SM[1](·)
SM[1](·)
SM[1](·)
SM[2](·)
SM[2](·)
SM[2](·)
$(\underline{\tilde{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LH}})^{\mathrm{T}}$
$(\underline{\tilde{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HL}})^{\mathrm{T}}$
$(\underline{\tilde{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{HH}})^{\mathrm{T}}$
$(\underline{\tilde{\mathbf{K}}}^{\mathrm{OH}}_{\mathrm{LL}})^{\mathrm{T}}$
$\hat{\mathbf{x}}$

Figure 4.5. High-level diagram of DHSN1 and DHSN2, in which the OHWT transform (encoding section) decomposes a noisy images into 4 sub-bands LL, LH, HL and HH, from which the high-passed bands (LH, HL and HH) are fed to a shared shrinkage module SM$[l](\cdot)$ which performs noise reduction and later feds the output to the inverse OHWT (decoding section). Here index l denotes the decomposition level. It should be noticed that the main difference between the original DHSN1 and the DHSN2 are the shrinkage module SM$[l](\cdot)$.

4.3.4 Shrinkage modules employed in the WSN

There are diverse existing denoising algorithms, such as the shrinkage estimators described in Chapter 3. In contrast, this section explores concepts based on the ideas of the method referred to as NeighShrink [54], where instead of a shrinkage function, a neighborhood-dependent multiplicative weight $\mathrm{W}(\cdot)$, is applied to each element of the high-pass bands of the OHWT $\underline{\mathbf{d}}$, where $\underline{\mathbf{D}} \in [f_{(2\downarrow)}(\mathbf{x}\circledast\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{LH}}),\ f_{(2\downarrow)}(\mathbf{x}\circledast\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HL}}),\ f_{(2\downarrow)}(\mathbf{x}\circledast\underline{\mathbf{K}}^{\mathrm{OH}}_{\mathrm{HH}})]$. The described concept can be mathematically defined by

$$\hat{\underline{\mathbf{D}}} = \mathrm{W}(\underline{\mathbf{D}}) \odot \underline{\mathbf{D}}. \tag{4.17}$$

Here, if the noise power is high with respect to the signal over a specific neighborhood in $\underline{\mathbf{D}}$, then $\mathrm{W}(\cdot)$ saturates to zero, thereby shrinking the noisy sample. In a converse reasoning, it can happen that an edge or detail of the signal occurs in which case $\mathrm{W}(\cdot)$ saturates to unity, thereby preserving the sample. It should be noticed that the neighbor dependency of NeighShrink sets it apart from most shrinkage-based algorithms, because most of them do not account for the correlation between pixels.

Despite the good performance of Neighshrink, its design assumes Gaussianity, which is not representative for the noise observed in CT. More importantly, in this work the low-contrast areas and texture should be preserved as much as possible. Therefore, we propose a shrinkage based on a data-driven weighting network that emulates the value of the weighting function $\mathrm{W}(\cdot)$ in Neighshrink. Specifically, this chapter explores two solutions. First, a heuristic design that learns the weighting function and second, a weighting structure that is more akin to the original NeighShrink, aiming at improving transparency by preserving the interpretability of the model. Both of these networks will be defined in more detail in the succeeding paragraphs.

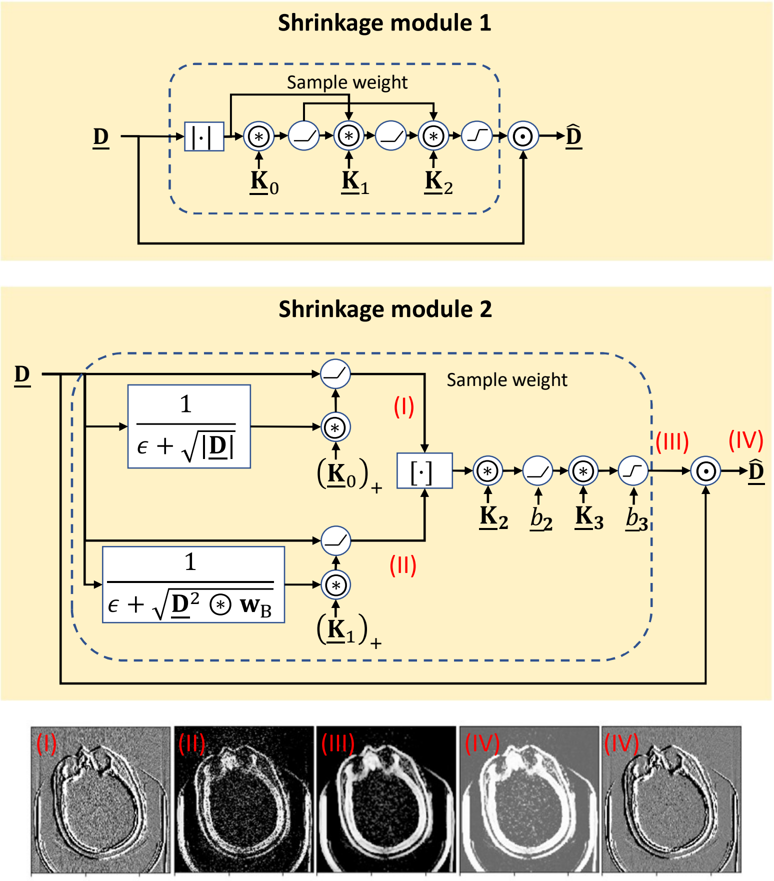

Shrinkage module 1
Sample weight
$\underline{\mathbf{D}}$
$|\cdot|$
$\underline{\mathbf{K}}_0$
$\underline{\mathbf{K}}_1$
$\underline{\mathbf{K}}_2$
$\hat{\underline{\mathbf{D}}}$
Shrinkage module 2
$\underline{\mathbf{D}}$
$\frac{1}{\epsilon + \sqrt{|\underline{\mathbf{D}}|}}$
$(\underline{\mathbf{K}}_0)_+$
Sample weight
(I)
$[\cdot]$
$\underline{\mathbf{K}}_2$
$\underline{b}_2$
$\underline{\mathbf{K}}_3$
$\underline{b}_3$
(III)
(IV)
$\hat{\underline{\mathbf{D}}}$
(II)
$\frac{1}{\epsilon + \sqrt{\underline{\mathbf{D}}^2 \circledast \mathbf{w}_\mathrm{B}}}$
$(\underline{\mathbf{K}}_1)_+$
(I)
(II)
(III)
(IV)
(IV)

Figure 4.6. Proposed shrinkage modules $\mathrm{SM1}(\cdot)$ (top) and $\mathrm{SM2}(\cdot)$ (middle). The middle row of pictures depict internal signals of stages within Module 2. Specifically, The outputs of the intermediate shrinkage layers are shown in Subfigures (I) and (II). The final generated mask used to suppress noise is displayed in Subfigure (III) and the noise-reduced band $\hat{\underline{\mathbf{D}}}$ is depicted in Subfigure (IV). It can be observed in signals (I) and (II) that the edge information is assigned a higher weight (close to unity), whereas the areas with little change are assigned a weight closer to zero. This is further visible when comparing the input and output signals (I) and (IV), respectively.

Module 1 For the first shrinkage module design, the multiplicative factor $\mathrm{W}(\cdot)$ is replaced by a CNN $\mathrm{SM1}(\cdot)$. For SM1, the design is guided by the following aspects. First, the shrinkage is only based on the amplitude and it uses the magnitude of the wavelet coefficients as input. Second, a convolutional layer generates linear combinations of the channels in the same band. Third, a ReLU operation is applied, which should saturate to zero in the presence of noisy components, providing a rough detail mask, which is then later refined by a residual connection. This concept can be repeated to improve the network operation. Fourth, the last layer saturates the output value between zero and unity with a sigmoid function, thereby suppressing noisy components, while preserving clean samples. A block diagram of the weighting network $\mathrm{W}(\cdot)$ is depicted in Fig. (4.6).

Module 2 The second shrinkage module ($\mathrm{SM2}(\cdot)$), further leverages ideas behind BayesShrink [14] and NeighShrink [54], which use the relative amplitude between the estimated noise variance and the signal power as features to perform wavelet shrinkage. In the case of BayesShrink, the signal power is globally taken into account, since the detail information is considered to have a Laplacian distribution, whereas the noise is assumed to be Gaussian and the threshold defines an optimal level that separates both distributions. In contrast, the empirical method NeighShrink generates an element-wise weight, based on a ratio between the local signal power against the global power of the noise. We find NeighShrink to be particularly interesting, because the calculations of the weights $\mathrm{W}(\cdot)$ are very similar to conventional operations used in CNNs. This becomes clear from the expression of NeighShrink [54] for which $\mathrm{W}(\cdot)$ is computed by

$$\mathrm{W}(\mathbf{d}) = \left(1 - \left[\lambda^2(\mathbf{d})\right] \cdot \frac{1}{\left[(\mathrm{Q}(\mathbf{d}))^2\right]}\right)_+, \tag{4.18}$$

where $\mathbf{d}$ is the input detail band, parameter λ is the universal threshold [60], which is proportional to the standard deviation of the noise. Furthermore, the function $\mathrm{Q}(\cdot)$ is the local standard deviation, defined by

$$\mathrm{Q}(\mathbf{d}[h,v]) = \frac{1}{n}\sqrt{\sum_{h=-\lfloor n/2 \rfloor}^{+\lfloor n/2 \rfloor} \sum_{v=-\lfloor n/2 \rfloor}^{+\lfloor n/2 \rfloor} \mathbf{d}^2[h,v]}, \tag{4.19}$$

where $\lfloor \cdot \rfloor$ is floor operator, truncating the number to the nearest integer with a lower absolute value and parameter n is the neighborhood chosen as an odd number. Finally, integer variables h, v are the horizontal and vertical indexes, respectively.

Motivated by the local nature of NeighShrink, we define a trainable shrinkage layer, of which the input feature is the inverse of the standard deviation of the noise within a neighborhood $\mathrm{Q}(\cdot)$, and where the universal threshold λ is replaced by a convolution kernel $\underline{\mathbf{C}}$. This shrinkage layer is then defined by the following expression:

$$\mathrm{F}_s(d) = \left(1 - \left[(\underline{\mathbf{C}})_+\right] \circledast \frac{1}{\left[\mathrm{Q}(\underline{\mathbf{D}}) + \epsilon\right]}\right)_+. \tag{4.20}$$

It should be noticed that Eq. (4.20) is an extended version of Eq. (4.18). We consider that outcome $\mathrm{F_s}(\underline{\mathbf{d}})$ produces a shrinkage estimate that can be used as intermediate feature in our noise reduction CNN. We hypothesize that the use of a convolution kernel $\underline{\mathbf{C}}$ may capture additional information provided by adjacent feature maps

$\mathbf{d} \in \underline{\mathbf{D}}$, rather than a global measure using a universal threshold λ. To ensure stability, $\underline{\mathbf{C}}$ must be non-negative which is enforced by using the ReLU operation. The constant ϵ is added for numerical stability.

In the module SM2$(\cdot)$, two different neighborhoods of 1 and 3 pixels are employed, where these signals serve as inputs to two independent instances of Eq. (4.20), each of which producing estimates A and B. The dimensions of the convolution kernel $\underline{\mathbf{C}}$ for A and B are $(N_{\mathrm{F}} \times N_{\mathrm{F}} \times 1 \times 1)$, where N_{F} is the number of feature maps of the band to which denoising is applied. After the shrinkage stage, this output is supplied to a two-layer perceptron that produces the final weighting, for which the input is the concatenation of the estimates A and B. The dimensions of the convolution kernels are $(2 \cdot N_{\mathrm{F}} \times 2 \cdot N_{\mathrm{F}} \times 1 \times 1)$ and $(N_{\mathrm{F}} \times 2 \cdot N_{\mathrm{F}} \times 1 \times 1)$, respectively, plus their respective biases. The first layer of the perception is followed by a ReLU activation and the second by a sigmoid $\sigma(\cdot)$ which ensures that the weight ranges between zero and unity, thereby enforcing that the network only can shrink detail components and not boost them. A full picture of the full shrinkage sub-network is depicted in Fig. 4.6.

An example of the operation of the proposed data-driven shrinkage network for the OHWT is presented in Fig. 4.6, where the meaning of the individual diagram steps are explained in the figure caption. The diagram is divided in three main stages (indicated at the top). The first stage is in charge of computing the local standard deviation estimate and its inverse, which are later fed to the second stage, where a shrinkage layer inspired by NeighShrink produces intermediate features. Finally, the last stage joins both of the NeighShrink estimates and produces a denoised estimate via an element-wise multiplication between the input band and the weights produced by the CNN.

To finalize the discussion on the shrinkage stage, the proposed design of SM1$(\cdot)$ and SM2$(\cdot)$ share the same data-driven shrinkage network across different bands at the same decomposition level, to reduce the number of parameters. The parameter reduction has two main advantages. First, it makes the models less prone to overfitting. Second, it reduces the memory requirements to train and execute the model.

4.4 Experiments and results

The experimental section of this chapter is divided in the following sections. First, Section 4.4.1 describes the dataset used for the experiments, the performance metrics employed as well as the reference CNNs. Afterwards, in Section 4.4.2 the experiments related to supervised noise reduction are addressed, while Section 4.4.3 discusses the results of the presented unsupervised noise reduction application.

The content for the present chapter is primarily sourced from the article *Image noise reduction based on a fixed wavelet frame and CNNs applied to CT* [57]. In the original, a few additional applications are presented that been excluded from this dissertation. The reason for this omission is to improve the storyline of the thesis and the integration with the following chapters. The excluded experiments are as follows. (1) An application of the models explored here as a regularizers/proximal operators in reconstruction in computed tomography. (2) Ablation studies of the proposed architecture highlighting the impact of the components of the shrinkage module of the DHSN2.

4.4.1 Data, performance metrics and reference CNNs

Section 4.4.1-A describes the dataset used for the experiments, while Section 4.4.1-B addresses the loss terms used for the (un)supervised training regimes used in this chapter, as well as some of the quality metrics used to evaluate the image estimates. Finally, Section 4.4.1-C addresses the reference CNNs used in the experiments.

A Dataset

For the experiments in this paper, we have used the 50 brain CT images from the cancer imaging archive [133]. For testing, 25 scans are applied in which at least one lesion was specified in the clinical data provided with the scans. The training and testing sets are shown in Table 4.1. We include the scan numbers for reference, because the clinical information provided with the dataset contains information regarding the image content and/or clinical conditions on each of the patients. Specifically, in the case of the scans selected for the test set, on patient basis the location of the lesion and clinical prognosis are provided.

Set	Scans
Training	N012, N024, N030, N047, N053, N072, N076, N082, N085, N113, N132, N139, N175, N176, N180, N187, N204, N209, N216, N236, N252, N264, N270, N293, N298
Testing	N005, N051, N056, N079, N090, N100, N105, N127, N138, N140, N141, N153, N156, N160, N169, N177, N181, N188, N190, N198, N202, N265, N279, N284, N300

Table 4.1. Training and testing sets used for the experiments. Here, the numbers indicate subject subsets.

B Training loss and image similarity metrics

This chapter explores two main applications for noise reduction, which are a supervised noise reduction in which the loss term $\mathcal{L}_{\text{sup}}$ used for training the CNN is defined by

$$\mathcal{L}_{\text{sup}}(\mathbf{y}, \mathbf{x}) = ||G(\mathbf{y}) - \mathbf{x}||_1. \tag{4.21}$$

Here, $G(\cdot)$ is the network being trained, while $\mathbf{y}$ and $\mathbf{x}$ are the noisy input and the noiseless ground truth, respectively. Finally, function $||\cdot||_1$ refers to the L_1 metric, that evaluates the distance between the image estimates produced by $G(\cdot)$ and the ground truth $\mathbf{x}$.

In addition to the supervised approach, an experiment is also performed in which we train the proposed WSN with an unsupervised scheme. Here, the model is exclusively based on noisy data (instead of using paired data to train the WSN to perform noise reduction). For this application, during training, the full-dose CT image in the pipeline is ignored. This resembles the deep-image prior (DIP) by Ulyanov *et al.* [134], where it is proposed to use noise reduction by training CNNs on noisy images and the training procedure stops before the CNN learns to represent the high-frequency details, which are assumed to be noise. In the proposed models in this chapter, the encoding-decoding path is a wavelet frame, that has perfect reconstruction. Therefore, we use a regularization term to ensure image smoothness, which is more akin to Oter *et al.* [135], where they exploit the total variation norm (TVN) together with the DIP framework.

Instead of using the TVN, we propose to improve the smoothness of QDCT scans by enforcing sparsity of the detail bands. This idea has been exploited earlier by many well-established wavelet-based noise reduction methods [14], [76]. The proposed approach maximizes the sparsity of the signals after the shrinkage stage. As a regularization parameter that ensures fidelity to the original, we adopt a similarity metric $M(\cdot)$ between the noisy input $\mathbf{x}$ against the processed image $G(\mathbf{x})$. The resulting loss term $\mathcal{L}_{\text{unsup}}$ is specified by

$$\mathcal{L}_{\text{unsup}}(\mathbf{x}) = \lambda \cdot \|G(\mathbf{x}) - \mathbf{x}\|_1 + \mathcal{R}(\mathbf{x}), \tag{4.22}$$

in which the constant λ defines a trade-off between signal sparsity and fidelity. Regularizer/proximal operator $\mathcal{R}(\mathbf{x})$ is defined by

$$\mathcal{R}(\mathbf{x}) = \frac{1}{L}\sum_{l=0}^{L-1}\Bigg(1 - \text{GI}\big(\text{SM}(\underline{\mathbf{d}}_l)\big)\Bigg), \tag{4.23}$$

in which $\underline{\mathbf{d}}_l$ are the samples of the high-pass bands of the wavelet-decomposed image $\mathbf{x}$ at the l-th level and $\text{GI}(\cdot)$ represents the Gini Index (GI) [136]. The GI is a well-known measure of sparsity and it is defined by

$$\text{GI}(\underline{f}) = 1 - 2\sum_{k=1}^{N}\frac{|\underline{f}_s[k]|}{||\underline{f}||_1}\cdot\left(\frac{N-k+\frac{1}{2}}{N}\right), \tag{4.24}$$

where $||\underline{f}||_1$ denotes the L_1 norm of $\underline{f}$, the vector $\underline{f} = (f[1], ..., f[N])$ is the input and $\underline{f}_s$ is the same vector, but with its elements sorted in an ascending magnitude. This description concludes the loss-function specification.

In order to assess the quality of the processed images, we use the mean structural similarity index metric [95] (MSSIM) between the processed QDCT and the FDCT scans (computed with Scipy [96] with the default parameters) and the peak signal-to-noise ratio (PSNR)), defined by Chapter 2, in which we define as maximum value MAX $= 80$ [HU], which is often used as the maximum intensity used in the

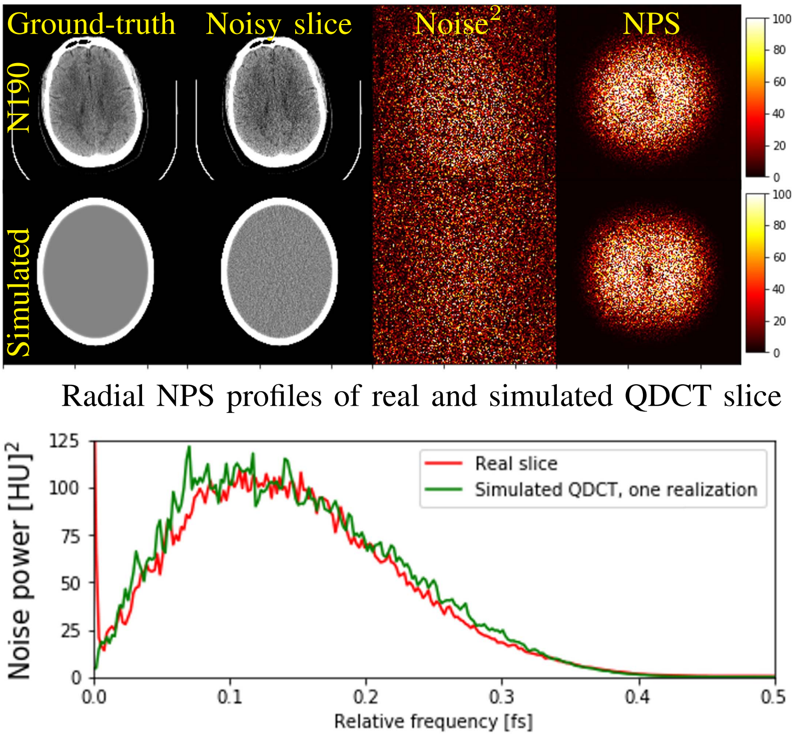


Figure 4.7. Noise Power Spectrum (NPS) of a slice of the dataset (upper row) and of a simulated phantom with one noise realization equivalent to one quarter dose (middle row), as well as a radial profile of the noise (lower row). The radial profiles of the QDCT slice of the dataset and of the simulated image are highly similar. Moreover, in both profiles the response around the DC frequency is close to zero, which should be the case because in the FBP reconstruction the signal is high-pass filtered with a ramp filter.

visualization of the brain parenchyma in CT. Furthermore, we have found that this value gives a good sensitivity to the PSNR reported here.

Noise in reconstructed CT images is non-stationary and non-Gaussian, due to the weighting, filtering and radial averaging of the sinogram during back-projection. In order to better characterize the noise effect introduced by processing and filtering stages –such noise reduction–, the noise power spectrum (NPS) is often used, since it allows to characterize the noise power at every spatial frequency [137]. Following Li *et al.* [138], the NPS is defined as

$$\mathrm{NPS}(\mathbf{x},\mathbf{y}) = \frac{1}{N_h}\frac{1}{N_v}\frac{\sum_{n=1}^{N_\mathrm{r}} ||\mathrm{DFT}_{\mathrm{2D}}(\mathbf{x_n}-\mathbf{y})||^2}{N_\mathrm{r}}, \tag{4.25}$$

where variables $\mathbf{x}_n$ and $\mathbf{y}$ are the n-th noisy input and the processed/ground-truth images, respectively. Furthermore, N_h and N_v are the number of pixels in the horizontal and vertical dimensions, while constant N_r is the number of noise realizations used to smoothen the NPS estimate and the function $\mathrm{DFT}_{\mathrm{2D}}(\cdot)$ denotes the two-dimensional

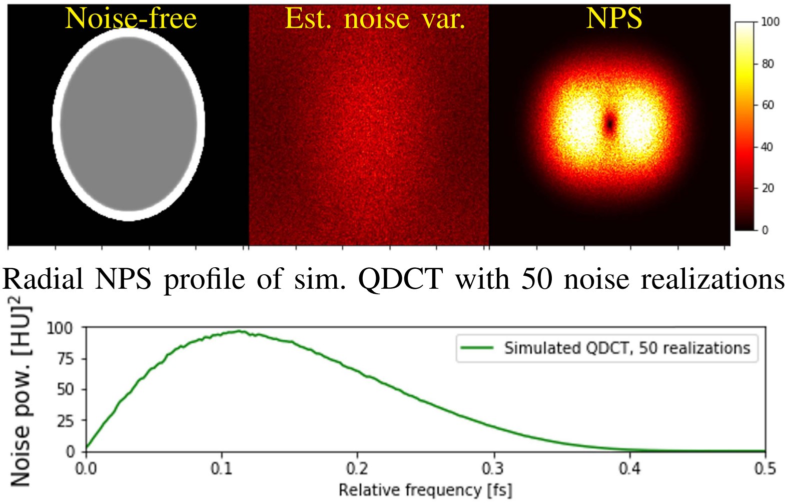


Figure 4.8. Pixel-wise noise variance, Noise Power Spectrum and its radial profile. For this figure, 50 noise realizations were averaged as indicated in Eq. (4.25).

discrete Fourier transform. Finally, complementary to the NPS, we obtain the radial profiles[2], which indicate the noise power as a function of the direction-independent spatial frequency.

For obtaining the NPS and its respective radial profile, we have employed a synthetic phantom composed of an oval shape with intensity in the range of bone and which is filled with a virtual material with an intensity similar to the gray matter in a brain CT scan. Additionally, noise has been added with a special frequency reminiscent to the noise present on the a slice of the QDCT of the subject N190. The noise signal is obtained by empirically adding Poisson-distributed noise to the sinogram of the synthetic phantom. The generated phantom and its noisy version, a reference FDCT and its simulated QDCT from subject N190, as well as their NPS and radial profiles are all visualized in Fig. 4.7. Furthermore, Fig. 4.8 depicts the noise variance per pixel estimated after 50 noise realizations, as well as its estimated NPS and its radial profile. Notice that the NPS has a *donut*-like shape due to the employed reconstruction kernel and due to the ramp filter used in filtered back-projection, which suppresses the zero frequency of the noise.

C Reference CNNs

As reference methods, we have adopted the residual encoder-decoder (RED) [3], which has been applied to noise reduction in reduced-dose CT, and the filtered back-projection network (FBPConvNet) [4], which is a multi-resolution CNN applied successfully to

[2] For the radial averages shown here, we have used the function *radial_profile* available at `https://stackoverflow.com/questions/21242011/most-efficient-way-to-calculate-radial-profile`.

[3] Code obtained from `https://github.com/SSinyu/RED-CNN`

[4] Adapted from `https://github.com/jvanvugt/pytorch-unet`

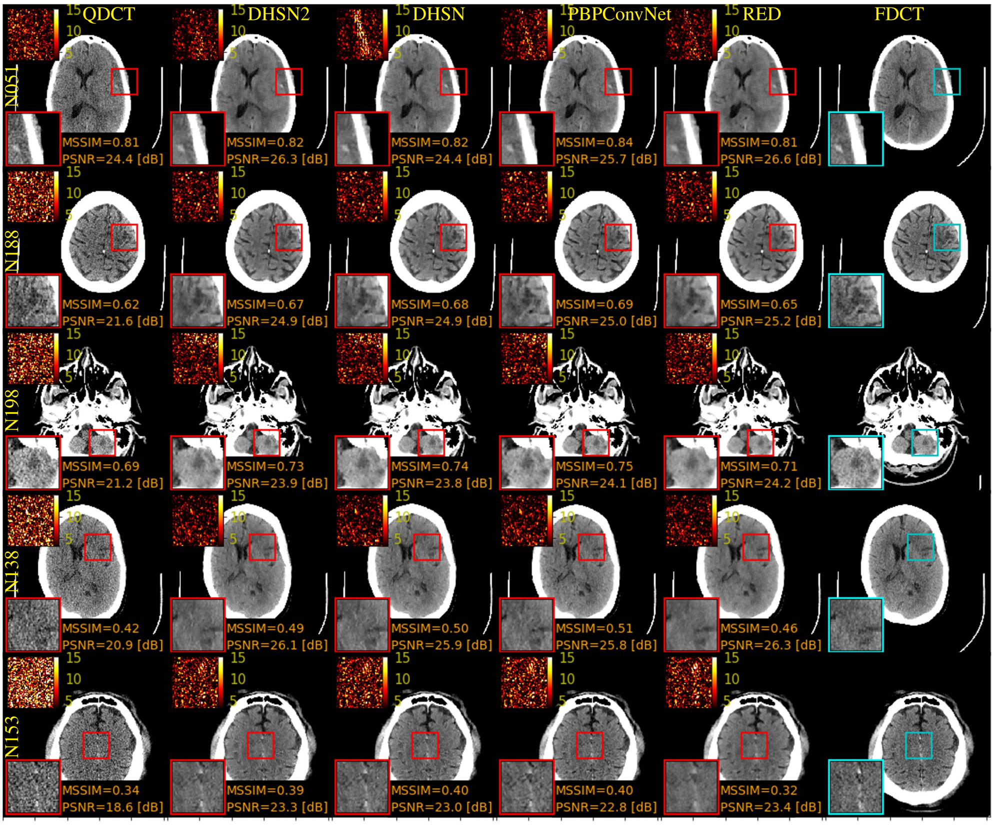

QDCT
DHSN2
DHSN
PBPConvNet
RED
FDCT
15
10
5
N051
MSSIM=0.81 PSNR=24.4 [dB]
MSSIM=0.82 PSNR=26.3 [dB]
MSSIM=0.82 PSNR=24.4 [dB]
MSSIM=0.84 PSNR=25.7 [dB]
MSSIM=0.81 PSNR=26.6 [dB]
N188
MSSIM=0.62 PSNR=21.6 [dB]
MSSIM=0.67 PSNR=24.9 [dB]
MSSIM=0.68 PSNR=24.9 [dB]
MSSIM=0.69 PSNR=25.0 [dB]
MSSIM=0.65 PSNR=25.2 [dB]
N198
MSSIM=0.69 PSNR=21.2 [dB]
MSSIM=0.73 PSNR=23.9 [dB]
MSSIM=0.74 PSNR=23.8 [dB]
MSSIM=0.75 PSNR=24.1 [dB]
MSSIM=0.71 PSNR=24.2 [dB]
N138
MSSIM=0.42 PSNR=20.9 [dB]
MSSIM=0.49 PSNR=26.1 [dB]
MSSIM=0.50 PSNR=25.9 [dB]
MSSIM=0.51 PSNR=25.8 [dB]
MSSIM=0.46 PSNR=26.3 [dB]
N153
MSSIM=0.34 PSNR=18.6 [dB]
MSSIM=0.39 PSNR=23.3 [dB]
MSSIM=0.40 PSNR=23.0 [dB]
MSSIM=0.40 PSNR=22.8 [dB]
MSSIM=0.32 PSNR=23.4 [dB]

Figure 4.9. Example slices of images in the test set highlighting lesions for the different methods under analysis. The displayed lesions are: Subject N051 - acute subdural hematoma, Subject N188 - chronic frontal infarct, Subject N198 - acute cerebellar infarcts, Subject N138 - acute frontal infarct, Subject N153 - traumatic subarachnoid hemorrhage. In the figure, the right column is the reference full-dose CT (FDCT) image. The leftmost column is the noisy quarter-dose CT (QDCT) input.

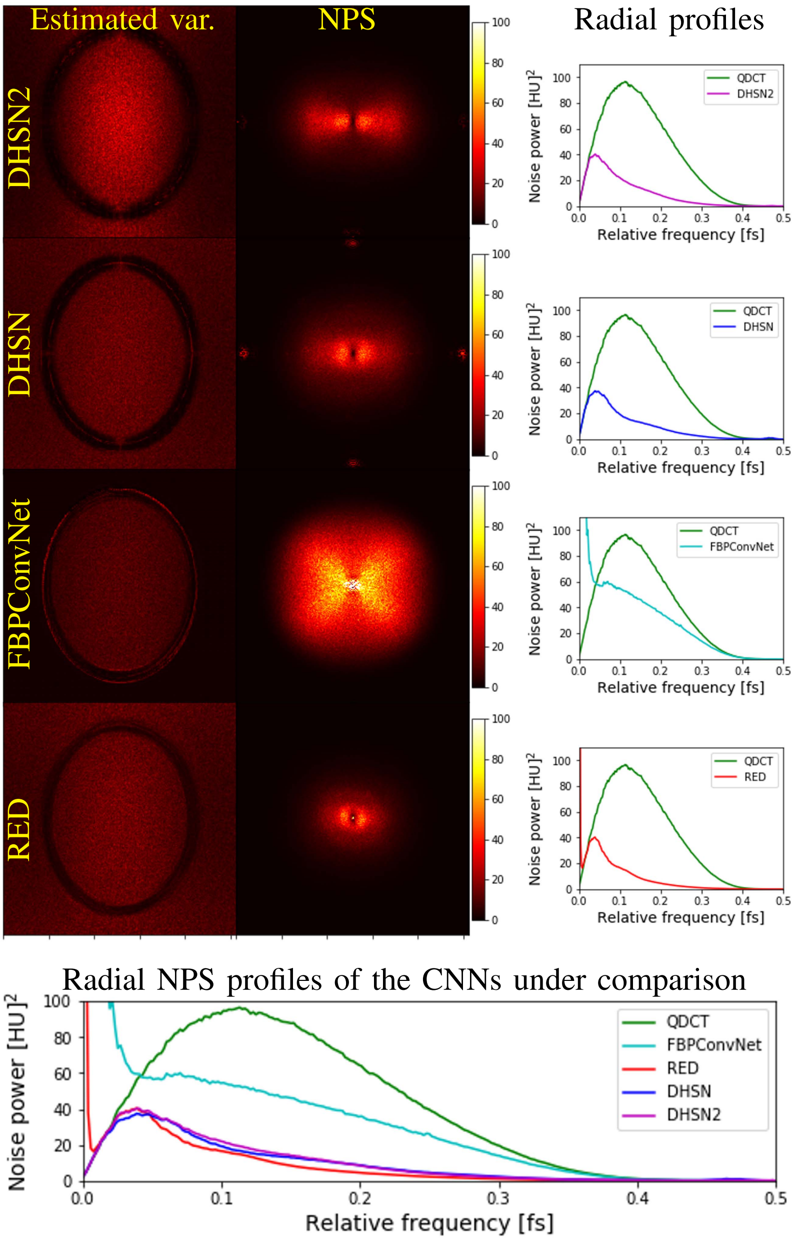


Figure 4.10. Estimated variances, noise power spectra (NPS) and radial profiles for the noise estimates produced by the DHSN2, DHSN1, FPBConNet and RED CNNs. The bottom graph integrates the NPS profiles of all the CNNs for better comparison.

CNN type	Total no. of parameters
DHSN2	35,196
DHSN	26,544
FBPConvNet	1,157,009
RED	465,889

Table 4.2. Number of parameters for the evaluated CNNs in the noise reduction experiments.

Model	MSSIM [AU]	PSNR [dB]
Baseline (QDCT)	0.650 ± 0.036	23.88 ± 1.83
FBPConvNet	0.657 ± 0.032	24.66 ± 1.64
RED	0.662 ± 0.031	$\mathbf{26.77} \pm \mathbf{1.86}$
DHSN1	0.662 ± 0.031	25.54 ± 1.68
DHSN2	$\mathbf{0.667} \pm \mathbf{0.031}$	26.20 ± 1.95

Table 4.3. Noise reduction performance for the supervised noise reduction in CT scans. The measured performance metrics are the MMSIM and PSNR measured in the test set. Here, AU stands for arbitrary unit, while the MSSIM values range between zero and unity.

suppression of artifacts in sparse-view CT. The number of trainable parameters of the networks under test are summarized in Table (4.2). In the case of the FBPConvNet, which is the closest to the proposed architecture, we approximately matched the number of feature maps and decomposition levels of the DHSN2. In addition, the implementation of all the networks shown here is available in GitHub [5].

4.4.2 Supervised noise reduction in CT images, generalization and phantom experiments

A Supervised noise reduction

For this experiment, the proposed WSN is trained with the Adam optimizer with a learning rate of 3×10^{-4} for 50 epochs with the QDCT and FDCT image pairs for each of the training sets, as shown in Table 4.1. Furthermore, a multiplicative learning-rate decay of 0.5 is applied every 7 epochs.

The SNR and MSSIM measurements of the QDCT images processed over the test set are shown in Table 4.3. Furthermore, as additional supporting evidence, Fig. 4.9 shows slices of the input and processed QDCT images obtained with each of the CNNs, as well as the ground-truth FDCT. In the figure, some of the lesions contained in the images in the images are highlighted and their SNR and MMSIM values are

[5] The code that replicates the most important results of this model is available at `https://github.com/LuisAlbertZM/Image-noise-reduction-based-on-a-fixed-wavelet-frame-and-CNNs-applied-to-CT`.

shown for the displayed patches. The depicted lesions are chosen to be of low-contrast nature (e.g. strokes or hemorrhages) to emphasize the preservation of information by each of the evaluated CNNs.

From Fig. 4.9 it is visible that the four networks under analysis are capable of achieving good noise reduction performance, which is backed up by the measurements shown in Table 4.3. Whereas the performance of the DHSN2, FBPConvNet and RED are comparable in terms of PSNR and MSSIM, it can be observed that the global metrics of Table 4.3 indicate that in terms of MSSIM, DHSN2 performs best, followed very closely by the original DHSN1 and RED. In contrast, when looking at the PSNR values, RED seems to perform best followed by the proposed DHSN2 and the original DHSN1. Furthermore, when looking at the MSSIM of the local metrics of the patches portrayed by Fig. 4.9, it can be noticed that RED generates the smoothest estimates, which favours its PSNR, but penalizes the MSSIM, as it is often the lowest-performing CNN in terms of the that metric.

The differences between the MSSIM and PSNR metrics shown in Table 4.3 between the DHSN2 and the original DHSN1 can be better understood when observing subject N051 and N198 in Fig. 4.9. In the figure, the DHSN1 has a higher error in the vicinity of high-intensity structures, which penalizes its PSNR and MSSIM, but it performs marginally better than the updated DHSN2 in the patches that do not contain skull (N138 and N153). In contrast, the updated DHSN2 minimizes errors in the high-intensity regions of the phantom. It is worth mentioning that Fig. 4.10 shows that the original DHSN1 and DHSN2 only reduce the intensity of the noise in the high-frequency segment of the Fourier spectrum (the high-frequency bands of the OHWT), which is a consequence of the spectrum partitioning produced by the wavelet transformation.

As shown in the summary of the theory of deep convolutional framelets [40] shown in Section 3.4.1, the encoding-decoding path of noise reduction CNNs can often be understood as a set of learned wavelet frames (the convolution kernels), in which noise reduction is performed with the ReLU non-linearities in a process akin to low-rank approximation. In the specific case of RED and FBPConvNet, it can be observed that the NPS profiles displayed in Fig. 4.10 are reminiscent to the pattern of DHSN2, in which a fixed wavelet frame is used. This suggests that, as proposed by Ye *et al.*, the ED paths of FBConvNet and RED are wavelet frames in which noise reduction is performed. However, in the reference CNNs, either the ED path, or the processing may be sub-optimal, since in FBPConvNet and RED a DC component is clearly noticeable in the Fourier domain. This could be caused by offset and/or aliasing introduced by these networks. Furthermore, it seems that the frequency partitioning provided by the ED path of FBConvNet may cover less of the low-frequency segment of the spectrum when compared with the DHSN2, DHSN1 and RED, since it can be observed that in the reference CNNs, the noise power in the low-frequency region of the spectrum is higher when compared with the DHSN2. In the case of RED, it seems that the network manages to reduce more noise in the high-frequency bands than the proposed method and achieves a similar NPS in the low-frequency spectrum, except for the DC level, where RED introduces a component which is not present in the DHSN2.

Set	Patient scan numbers				
Subset 0	N012,	N024,	N030,	N047,	N053
Subset 1	N072,	N076,	N082,	N085,	N113
Subset 2	N132,	N139,	N175,	N176,	N180
Subset 3	N187,	N204,	N209,	N216,	N236
Subset 4	N252,	N264,	N270,	N293,	N298

Table 4.4. Partitions of patients in the training set, used for testing generalization.

	PSNR on test set [dB]				
Network	Subset 0	Subset 1	Subset 2	Subset 3	Subset 4
FBPConvNet	24.68	24.65	24.63	24.49	24.36
RED	26.71	26.69	26.76	26.68	26.31
DHSN1	25.92	25.43	26.04	25.71	25.68
DHSN2	26.22	26.16	26.22	26.24	26.13

Table 4.5. Measured PSNR values in the test set for the generalization experiment.

B Generalization of the analyzed networks

For testing the generalization of the considered CNNs, the training set is split into 5 subsets, which is motivated by assessing the consistency of the results when exposing the different models to a smaller sample of the training data. If a model generalizes well, then the influence on the quality of the estimates by changing the training set should be small. The selected subjects used for each of the subsets are shown in Table 4.4. In order to conduct a similar amount of training than in the previous experiment with the reduced training sets, we have increased the number of epochs to 250 and the period for the multiplicative learning-rate decay to 35 epochs. Akin to the previous experiment, the PSNR values are measured for each of the trained images which are summarized in Table 4.5. It is worth mentioning that the MSSIM has been excluded for this experiment because it has been found that the results of this metric are redundant with the PSNR.

From the data, it can be observed that the DHSN2 is only marginally affected by changing the subset of the training set for the CNN learning (the difference in PSNR between the best and worst model is only 0.11 dB). In contrast, the DHSN1 has larger variations between the best and worst performing models, i.e 0.61 dB. Finally, FBPConvNet and RED also behave consistently, although less than the proposed DHSN2 (with differences between best and worse models of 0.32 and 0.40 dB, respectively). Therefore, from this ablation experiment, it can be concluded that FBPConvNet is the mostly affected by changing the training dataset.

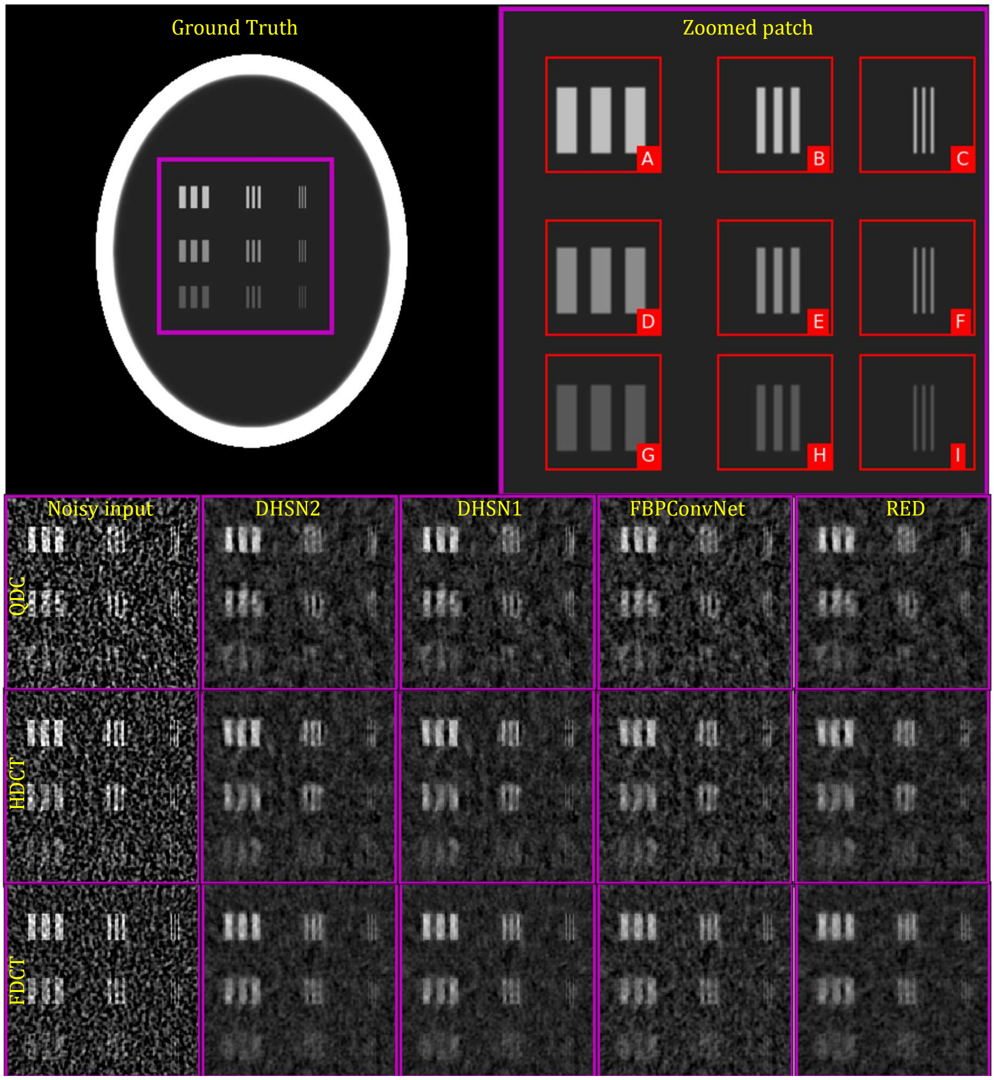


Figure 4.11. Noisy phantom and estimates produced by the tested CNNs for testing preservation of image content as function of contrast, width and dose. Due to the low-contrast nature of the patterns in the virtual phantom, the display range is set to 25 to 75 HU.

C Phantom experiments, test preservation of fine detail and CNN performances for different dose

To test the effect of the DHSN2, DHSN1, FBPConvNet and RED in terms of preservation of low-contrast and narrow image content, we have generated a test phantom with synthetic noise with a the same procedure as described in Section 4.4.1-B. In the phantom, a test pattern is inserted based on lines with 9 distinct Regions Of Interest (ROIs) A through I (see Fig. 4.4.1-A). The amplitude of patterns inside the selected

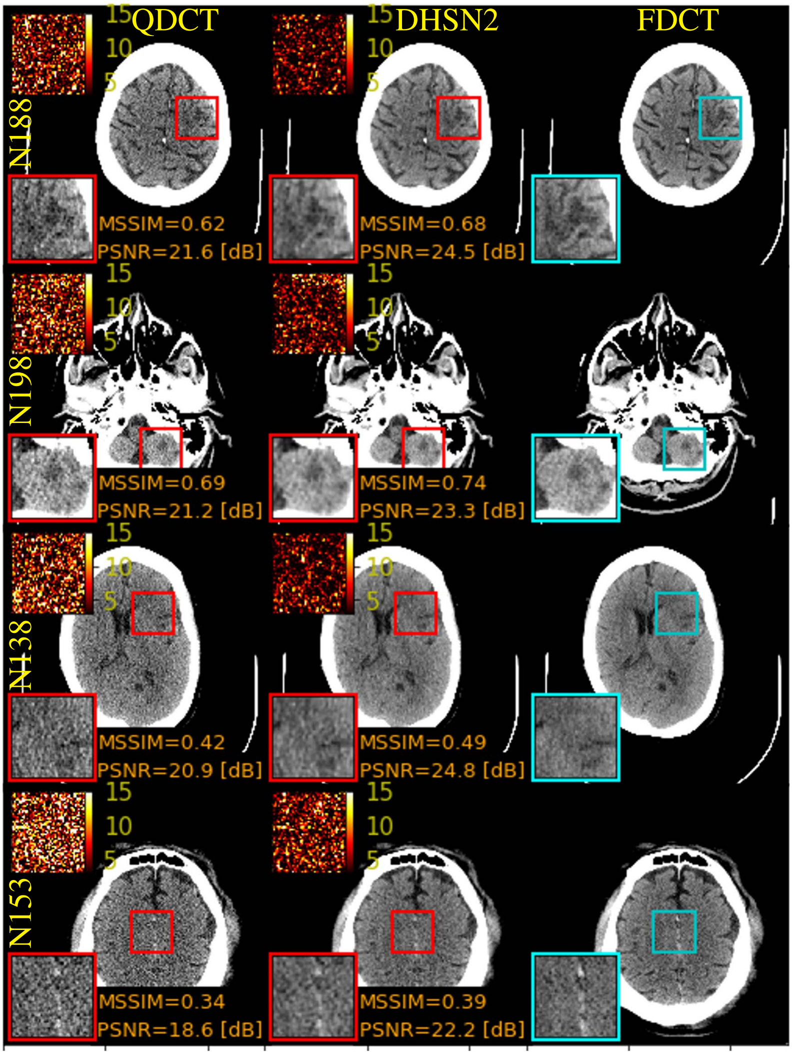


Figure 4.12. Examples of slices of scans filtered with the WSN2 trained without FDCT ground-truth. In the image is shown the reference QDCT, the processed QDCT with the DHSN2 and the reference FDCT.

ROIs are as follows. Patterns within [A, B, C] have amplitude 15 HU, the lines inside [D, E, F] 10 HU. Finally, the elements in [G, H, I] have amplitude of 5 HU. Furthermore, the patterns have distinct widths and spacing to test the spatial resolution of the signal estimates with the following characteristics. Patterns within [A, D, G] have width of 7 pixels and a separation of 5 pixels, [B, E, H] are 3 pixels wide with a spacing of 3 pixels. Finally, [C, F, I] are 1 pixel wide and are spaced 2 pixels.

Table 4.6 summarizes the MSSIM values for the estimates produced by the evaluated CNNs, measured on each of the ROIs A through I for the simulated quarter-dose

CT slice (QDCT). Furthermore, in Fig. 4.11, the estimates produced by the diverse CNNs are shown. As seen in Fig. 4.11, all the CNNs manage to recover the larger patterns, although the estimates degrade as the contrast and width decreases. For most of the selected ROIs, the best performing CNN is the DHSN1, while DHSN2 and FPBConvNet are usually in second place in terms of MSSIM.

To test the performance of the CNNs when processing images with different dose levels, we have simulated acquisitions of the phantom with half-dose and full-dose (referred to as HDCT and FDCT, respectively). The FDCT, HDCT and QDCT and the noise-reduced images are portrayed by Fig. 4.11. The MSSIM between the processed images and the noiseless ground truth are summarized in Table 4.7. The MSSIM values show that for all the dose levels, the DHSN1 performs best. In the case of the QDCT and HDCT images, the DHSN2 performs closely to the DHSN1. This suggests that the DHSN1 is slightly better in preserving the low-contrast information, while the DHSN2 is better at preserving global structures, as shown by the global measurements of Table 4.3. This may be caused by the NeighShrink-inspired layer, in which less weight is given to sections of the image with lower-intensity edges. This will be further investigated in future work. In the case of the FDCT of (Fig. 4.11), the input has little noise and the blur introduced by DHSN2, FBPConvNet and RED penalizes the MSSIM of the processed images as shown in Table 4.7.

ROI	Input	DHSN2	DHSN1	FBPConvNet	RED
A	0.623	**0.661**	**0.679**	0.648	0.629
B	0.491	**0.593**	**0.604**	0.583	0.553
C	0.261	**0.429**	**0.444**	0.417	0.390
D	0.366	0.252	**0.295**	**0.273**	0.253
E	0.312	0.333	**0.357**	**0.335**	0.307
F	0.136	0.196	**0.208**	0.188	**0.205**
G	0.217	0.172	**0.225**	**0.178**	0.158
H	0.185	0.165	**0.208**	**0.180**	0.161
I	0.053	**0.067**	**0.085**	0.066	**0.067**

Table 4.6. MSSIM for ROIs with varying contrast and thickness measured in processed QDCT.

Dose	Input	DHSN2	DHSN1	FBPConvNet	RED
QDCT	0.117	**0.130**	**0.140**	0.129	0.123
HDCT	0.132	**0.135**	**0.146**	**0.135**	0.127
FDCT	**0.147**	0.137	**0.151**	0.139	0.131

Table 4.7. MSSIM for synthetic patch as function of the dose level.

4.4.3 Unsupervised noise reduction application

For this experiment, the DHSN2 is trained with the procedure discussed in Eq. (4.22). The training of the DHSN2 is performed with the Adam optimizer with a learning rate of $1{\times}10^{-3}$ for 12 epochs with the QDCT images of the training set of 25 scans, as shown in Table 4.1, while applying a learning-rate decay of 0.5 every 4 epochs. The regularization parameter λ of Eq. (4.22) is set to 0.0135. With this procedure, we have obtained average MSSIM and and PSNR values equal to 0.655±0.032 and 24.63±1.36 dB, respectively, which is slightly superior to the QDCT data (0.650±0.036 as shown in Table 4.3). In addition, considering the individual patches and slices shown in Fig. 4.12, the DHSN2 manages to achieve a significant noise reduction in this modality. In particular, the PSNR is significantly higher than the QDCT references for all the slices shown. It should be noted that in the zoomed difference of subjects N198 and N051, the the gain in PSNR is slightly lower in the patches that contain larger parts of the skull. We have found that by allowing more regularization, the error in the bones increases and penalizes heavily on the PSNR, due to the high-intensity values of bones in HU.

4.5 Conclusions and future work

This chapter presents two designs of encoding-decoding CNNs in which the the encoder and decoder of the networks are fixed/non-trainable convolution filters. The used framelet basis is referred to as *overcomplete Haar wavelet transform* (OHWT), which is a special case of the dual-tree complex wavelet transform. Furthermore, this chapter presents also two CNN models that leverages the proposed OHWT to remove noise in CT images (DHSN1 and DHSN2). Specifically, these models employ the CNN model in the OHWT domain, which allows the CNNs to leverage the sparse representation and perfect reconstruction of the OHWT.

Contributions. The proposed framelet transform and shrinkage functions allow for good performing CNNs which offer performance similar to well-known denoising CNN methods, while preserving interpretability. Specifically, the convolution filters of the encoding and decoding paths are a linear transformation in which a shrinkage function is applied. This increases the certainty on what the model is doing. In summary, this approach presents the following contributions.

1. *Framelet representation*. This chapter proposes the *overcomplete Haar wavelet transform* as basis representation. This transform is a special case of the dual-tree complex wavelet transform and differs from the DTCWT algorithm in the following ways. (1) The proposed transform uses Haar basis functions as filters in all the decomposition levels. In addition, the phase shift between tree “a" and “b" is one full sample, rather than half a sample, as is done in the DTCWT. The proposed method produces a non-analytic representation with compact and directional filters. The proposed basis also can be implemented as a single convolutional filter, which avoids sequential operations and allows for a simple

integration within CNNs. (2) All the wavelet bands (LL, LH, HL, HH) use the proposed OHWT, which contrasts with the original DTCWT in which the additional processing steps are only applied to the the high-pass bands (LH, HL, HH). As a result, the OHWT is a more redundant transform in which every band represents a narrower frequency band, which leads to a sparser representation.

2. *Shrinkage function*. The presented shrinkage CNN combines neural networks and feature-based machine learning. That is, the proposed method actively exploits the wavelet representations where noise reduction is performed, where the noise reduction is based on the computation of intermediate features based on the local power of the detail coefficients.

The tight integration of deep learning and wavelet denoising within the proposed DHSN1 and DHSN2 CNNs generalizes well, avoid processing in the low-frequency band, while these models are easy to implement and they have an interpretable operation. It can be hypothesized that for image restoration tasks, the use of this type of architectures may offer advantages over conventional CNNs, because they focus the learning procedure on the noise reduction task, rather than also learning to decompose and reconstruct the signals like RED and FBPConvNet do.

The next chapter continues the exploration of designing noise reduction CNNs and explores an alternative approach in which the encoding-decoding path is learned, where it can be proven that the network can achieve approximate perfect reconstruction, both practically and theoretically.

CHAPTER 5

Data-driven denoising using trainable framelets

5.1 Introduction

The previous chapter has presented a CNN where the encoding and decoding filters are non-trainable tight framelets, while the trainable part of the CNN performs the noise reduction process. This design choice leads to a simple and interpretable model, because the signal decomposition, reconstruction and denoising are clearly defined in the design. This chapter explores an alternative model, where both the encoding-decoding and noise reduction steps are learned and in which it is verifiable that (a) the encoding and decoding paths behave akin to tight framelets and (b) the non-linear section suppresses the noise of the signal.

Challenges. As presented in Chapter 3, common CNNs (e.g. FBPConvNet [50]) contain highly non-linear paths where (almost) every convolution layer is followed by a non-linear activation. This configuration is powerful and it has a similar mathematical formulation to well-known signal processing techniques for noise reduction based on framelets and low-rank approximation. Still, the data-driven nature of CNNs, does not guarantee that CNNs converge to a set of parameters which ensure that the trained models behave exactly as low-rank approximators, as discussed in the text boxes of *Fitting low-rank approximation in ReLU CNNs* and *Network depth* (in Section 3.4) and, as well as Section 3.6. Consequently, in order to achieve an encoding-decoding path where the convolution filters are actual tight framelets, there are challenges that should be addressed. These challenges that are described now.

1. *Design of the encoding-decoding path*. In order to achieve a data-driven CNN where the encoder and decoder are tight framelets, the linear part of the encoding-decoding path should allow for perfect reconstruction, otherwise the design could not represent all the spatial frequencies evenly *by design* (e.g. the U-Net/FBPConvNet example shown in Section 3.5.1).
2. *Model complexity*. As shown in Section 3.3.3, the choice for non-linearity contains assumptions about and enforces properties to the feature maps of CNNs. For example, both the ReLU and soft-shrinkage functions enforce sparsity on the feature maps by setting low-amplitude samples to zero. Still, the ReLU function also enforces the signal to be positive. Consequently, if the feature maps

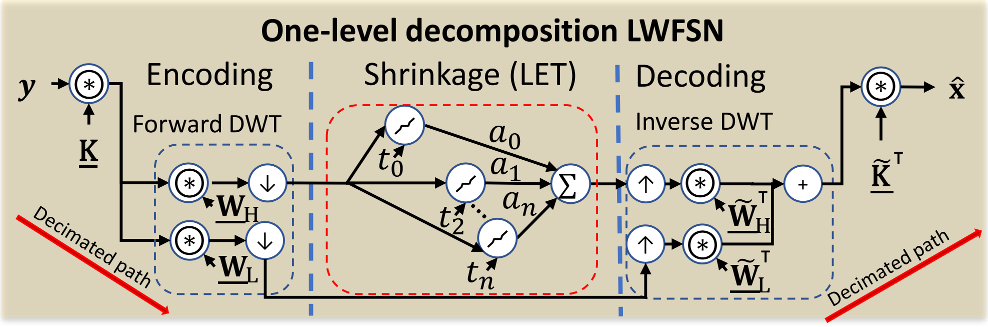

One-level decomposition LWFSN
Encoding
Shrinkage (LET)
Decoding
Forward DWT
Inverse DWT
y
$\underline{\mathbf{K}}$
$\underline{\mathbf{W}}_H$
$\underline{\mathbf{W}}_L$
t_0
t_2
t_n
a_0
a_1
a_n
Σ
$\underline{\tilde{\mathbf{W}}}_H^T$
$\underline{\tilde{\mathbf{W}}}_L^T$
$\underline{\tilde{\mathbf{K}}}^T$
$\hat{\mathbf{x}}$
Decimated path
Decimated path

Figure 5.1. One-level LWFSN architecture, in which it is visible that the LWFSN has clear encoding, denoising and decoding paths. In the figure, filter $\underline{\mathbf{W}}_{\mathrm{H}}$ represents the high-frequency filters of the DWT $\underline{\mathbf{W}}_{\mathrm{H}} = (\mathbf{w}_{\mathrm{LH}}, \mathbf{w}_{\mathrm{HL}}, \mathbf{w}_{\mathrm{HH}})$, while $\underline{\mathbf{W}}_{\mathrm{L}} = (\mathbf{w}_{\mathrm{LL}})$ is the filter of the low-frequency band. In addition, $\underline{\mathbf{K}}_l$ and $\underline{\tilde{\mathbf{K}}}_l$ are the learned convolution filters in the encoding-decoding paths, while variables t_n and a_n are the threshold values and weights used in the thresholding function implemented via the linear expansion of thresholds.

contain both positive and negative samples, additional channels are required to avoid signal loss (see Section 3.4.1). This increases the computational cost of training/deploying ReLU-based models, as well as the number of parameters, which consequently augments the risk of overfitting.

3. *Model interpretability*. The use of trainable filters as encoding-decoding path poses a challenge in model interpretability. For example, low-rank approximation and wavelet shrinkage can operate differently than the learned structure on the mechanisms of CNNs, in spite of their similar mathematical formulations. The models presented in this chapter should display properties that prove that the proposed models operate more closely to algorithms based on wavelet shrinkage.

Background/related work. Among the first models exploiting interpretations of CNNs-based wavelet shrinkage and/or low-rank approximation to improve the operation of encoding-decoding CNNs, we find the so-called filtered-backprojection network (FBPConvNet) [50] as well as the tight frame U-Net [36]. In the latter network, the down/up-sampling structure of the conventional U-Net is replaced by the discrete wavelet transform and its inverse. Consequently, the down/up sampling is composed by tight frames with perfect reconstruction, which improves its performance for artifact removal in compressed sensing imaging for CT. Still it should be noted that despite the down/up-sampling structure are tight frames, the same cannot be assumed for the encoding and decoding filters which are learned.

More active attempts to achieve encoding encoding paths with a predictable operation have appeared in literature. For example, the *convolutional analysis operator* [139], is a model where a regularization CNN is used to eliminate artifacts from CT images. Within this framework, the regularizer is trained to eliminate the artifacts, while ensuring that the encoder and decoder allow for perfect reconstruction. A similar approach for the regularizer of *FISTA-Net* [111], which s explicitly trained also to achieve perfect reconstruction and to provide a sparse representation for the data.

An alternative approach to improve the interpretability of noise reduction CNNs is to make that encoding-decoding structure and that the non-linear section of CNNs mimics the operation of more conventional framelet-based denoising designs (e.g. BayesShrink [14]). This can be achieved by simplifying the CNNs and to use activations that are more reminiscent to conventional noise reduction designs such as the soft-shrinkage operation. An example of such approaches is the *soft autoencoder* [39]. This network is a single-resolution CNN in which soft-shrinkage functions are placed as activations in the encoder, while the decoder of the CNN remains linear. Alternatively, the *multi-scale sparse coding network* [48] presents a solution in which single-layer CNNs present a single non-linearity placed between both paths.

Direction of the followed approach. In order to address the above challenges, this chapter proposes a model referred to as *learned wavelet frame shrinkage network* (LWFSN). This model has a linear encoding-decoding path, where the DWT is used as up/down-sampling structure. Furthermore, the model also contains the semi-hard thresholds as non-linearities in between the encoder and decoder.

The simple design of the LWFSN has the following advantages. (1) The use of semi-hard thresholds reduces the number of channels/feature maps of the model when compared with ReLU-based models. This decreases the number of trainable parameters and memory required to train/deploy the model. (2) The design is simpler and less redundant than alternative models. (3) The placement of the non-linearities in the middle of the encoder and decoder sections and the up/down-sampling path based on the DWT allows to leverage the DWT filters not only for down/up-sampling, but also to make sparser feature maps. This overcomes the limitations of the multi-scale sparse coding network discussed in Section 3.5.4. (4) The simple and almost linear design of the LWFSN eases to provide experimental evidence that the filters of the encoding and decoding path behave akin to tight framelets. Specifically for this purpose, the impulse response analysis over the linearized encoding-decoding path is employed. This practical analysis complements the theoretical approaches of Section 3.5 and increases the certainty on what the model is doing.

Structure of this chapter. The structure of this chapter is as follows. First, Section 5.2 introduces the design of the (residual) LWFSN as well as an analysis of its reconstruction characteristics. Afterwards, Section 5.3 presents the performance of the LWFSN for noise reduction as well as its impulse response. Section 5.4 discusses the the experimental results. Finally, Section 5.5 draws the conclusions on the proposed design.

5.2 Methods

5.2.1 Overview of the design of the LWFSN

As mentioned in Section 5.1, the main source of inspiration for the model presented in this chapter is the tight-frame (TF) U-Net [36]. Prior to the development of this work, Section 3.5 shows an exhaustive analysis of the design of the TF U-Net that shows simplifications which can be performed to this design. These aspects are listed as follows. First, the ReLUs can be replaced by semi-hard thresholding functions.

This modification reduces the redundancy required for perfect reconstruction and ensures that the CNN learns to perform wavelet thresholding. Second, the nonlinearities are placed in between the forward and inverse DWT, instead of within the encoding-decoding path. This is done with the purpose of leveraging the high-pass filters of the DWT, since they generate the zero-mean signals that are required by the shrinkage functions. Third, and based on Section 3.5, it is noticed that the number of trainable parameters can be further reduced by eliminating the skip connection(s) of the undecimated path.

Altering the TF U-Net with the proposed modifications generates the so-called learned wavelet frame shrinkage network (LWFSN). It is possible to prove that the encoding-decoding path of the proposed design can theoretically achieve an encoding-decoding with approximate perfect reconstruction. Furthermore, the LWFSN design has low memory footprint. This allows to use a wider variety of computing hardware for training/deployment and to process larger images and/or to train deeper networks.

The proposed LWFSN with one decomposition level is shown in Fig. 5.1 and is described mathematically by

$$\begin{aligned} L(\mathbf{y}) =& f_{(2\uparrow)}\left(\tau^{\mathrm{LET}}_{(\underline{t})}\left(f_{(2\downarrow)}\left(\mathbf{y}\circledast\underline{\mathbf{K}}\circledast\underline{\mathbf{W}}_{\mathrm{H}}\right)\right)\circledast\underline{\tilde{\mathbf{W}}}_{\mathrm{H}}^{\intercal}\circledast\underline{\tilde{\mathbf{K}}}^{\intercal}\right)+\\ & f_{(2\uparrow)}\left(f_{(2\downarrow)}\left(\mathbf{y}\circledast\underline{\mathbf{K}}\circledast\underline{\mathbf{W}}_{\mathrm{L}}\right)\right)\circledast\underline{\tilde{\mathbf{W}}}_{\mathrm{L}}^{\intercal}\circledast\underline{\tilde{\mathbf{K}}}^{\intercal}. \end{aligned} \tag{5.1}$$

Here, tensor $\underline{\mathbf{K}}$ is the learned convolution filter and $\underline{\tilde{\mathbf{K}}}$ is its (learned) inverse, function $\tau^{\mathrm{LET}}_{(\underline{t})}(\cdot)$ is some form of thresholding applied to the high-pass signal $\mathbf{y}\circledast\underline{\mathbf{K}}\circledast\underline{\mathbf{W}}_{\mathrm{H}}$, which in this chapter is implemented via the *linear expansion of thresholds* (LET) approach discussed in Section 3.3.3-D. In the following subsections, if deeper networks are required, this can be achieved by further decomposing the low-frequency signal $\mathbf{y}\circledast\underline{\mathbf{K}}\circledast\underline{\mathbf{W}}_{\mathrm{L}}$.

A Encoding-decoding path of the LWFSN

In Eq. (5.1) it can be noticed that there are no ReLUs in between the trainable convolution filter $\underline{\mathbf{K}}$ and the fixed convolution sub-filters of the DWT ($\underline{\mathbf{W}}_{\mathrm{L}}$ and $\underline{\mathbf{W}}_{\mathrm{H}}$). Therefore, we can rewrite Eq (5.1) into

$$\begin{aligned} L(\mathbf{y}) =& f_{(2\uparrow)}\left(\tau^{\mathrm{LET}}_{(\underline{t})}\left(f_{(2\downarrow)}\left(\mathbf{y}\circledast\underline{\mathbf{F}}_{\mathrm{H}}\right)\right)\circledast\underline{\tilde{\mathbf{F}}}_{\mathrm{H}}^{\intercal}\right)+\\ & f_{(2\uparrow)}\left(f_{(2\downarrow)}\left(\mathbf{x}\circledast\underline{\mathbf{F}}_{\mathrm{L}}\right)\right)\circledast\underline{\tilde{\mathbf{F}}}_{\mathrm{L}}^{\intercal}, \end{aligned} \tag{5.2}$$

in which the convolution filters $\underline{\mathbf{F}}_{\mathrm{H}} = \left(\underline{\mathbf{F}}_{\mathrm{LH}},\underline{\mathbf{F}}_{\mathrm{HL}},\underline{\mathbf{F}}_{\mathrm{HH}}\right)$ and $\underline{\mathbf{F}}_{\mathrm{L}} = \left(\underline{\mathbf{F}}_{\mathrm{LL}}\right)$ are parts of the frame $\underline{\mathbf{F}}$, defined by

$$\underline{\mathbf{F}} = \begin{pmatrix}\underline{\mathbf{F}}_{\mathrm{LL}}\\ \underline{\mathbf{F}}_{\mathrm{LH}}\\ \underline{\mathbf{F}}_{\mathrm{HL}}\\ \underline{\mathbf{F}}_{\mathrm{HH}}\end{pmatrix}^{\intercal} = \begin{pmatrix}\underline{\mathbf{K}}_0\circledast\underline{\mathbf{K}}_1\circledast\mathbf{w}_{\mathrm{LL}}\\ \underline{\mathbf{K}}_0\circledast\underline{\mathbf{K}}_1\circledast\mathbf{w}_{\mathrm{LH}}\\ \underline{\mathbf{K}}_0\circledast\underline{\mathbf{K}}_1\circledast\mathbf{w}_{\mathrm{HL}}\\ \underline{\mathbf{K}}_0\circledast\underline{\mathbf{K}}_1\circledast\mathbf{w}_{\mathrm{HH}}\end{pmatrix}^{\intercal}. \tag{5.3}$$

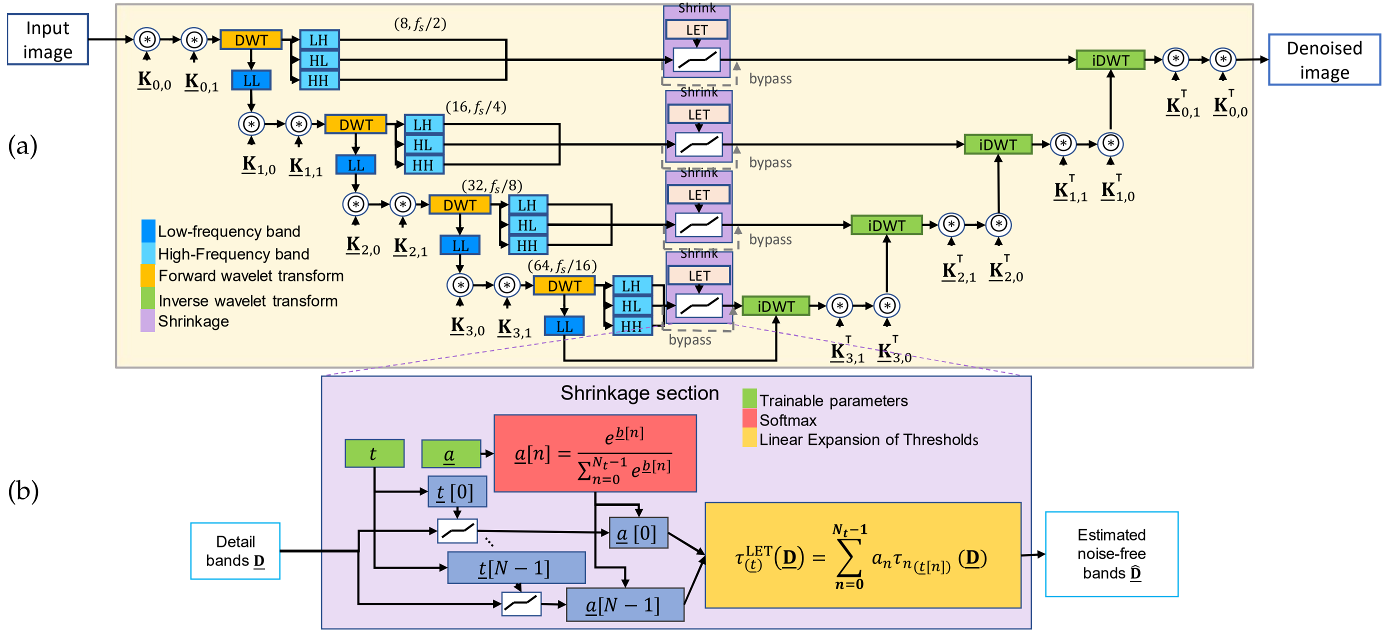

(a)
Input image
Denoised image
DWT
LL
LH
HL
HH
$(8, f_s/2)$
$(16, f_s/4)$
$(32, f_s/8)$
$(64, f_s/16)$
Shrink
LET
bypass
iDWT
$\mathbf{K}_{0,0}$ $\mathbf{K}_{0,1}$
$\mathbf{K}_{1,0}$ $\mathbf{K}_{1,1}$
$\mathbf{K}_{2,0}$ $\mathbf{K}_{2,1}$
$\mathbf{K}_{3,0}$ $\mathbf{K}_{3,1}$
$\mathbf{K}_{0,1}^{T}$ $\mathbf{K}_{0,0}^{T}$
$\mathbf{K}_{1,1}^{T}$ $\mathbf{K}_{1,0}^{T}$
$\mathbf{K}_{2,1}^{T}$ $\mathbf{K}_{2,0}^{T}$
$\mathbf{K}_{3,1}^{T}$ $\mathbf{K}_{3,0}^{T}$
Low-frequency band
High-Frequency band
Forward wavelet transform
Inverse wavelet transform
Shrinkage
(b)
Shrinkage section
Trainable parameters
Softmax
Linear Expansion of Thresholds
Detail bands $\underline{\mathbf{D}}$
t
a
$\underline{a}[n] = \frac{e^{\underline{b}[n]}}{\sum_{n=0}^{N_t-1} e^{\underline{b}[n]}}$
$\underline{t}\,[0]$
$\underline{t}[N-1]$
$\underline{a}\,[0]$
$\underline{a}[N-1]$
$\tau_{(\underline{t})}^{\mathrm{LET}}(\underline{\mathbf{D}}) = \sum_{n=0}^{N_t-1} a_n \tau_{n(\underline{t}[n])}(\underline{\mathbf{D}})$
Estimated noise-free bands $\hat{\underline{\mathbf{D}}}$

Figure 5.2. Proposed architecture LWFSN. The upper part (Subfigure (a)) is our proposed learned wavelet frame shrinkage network in which the encoding-decoding path is formed by a set of convolution with filters $\underline{\mathbf{K}}_{l,j}$ –where l denotes the decomposition level and j the tensor index–, the discrete wavelet transform and its inverse, denoted as DWT and iDWT, respectively. In the skip-connection path the shrinkage function is performed by a LET, detailed in Subfigure (b), in which the trainable parameters t and b are used as the threshold level and to compute the weight associated to each thresholding function, respectively. In the figure f_s the sampling frequency, therefore a f_s/n denotes a decimation of factor n.

Here, we define $\underline{\mathbf{F}}$ as the separable filter $\underline{\mathbf{F}} = \underline{\mathbf{K}}_0 \circledast \underline{\mathbf{K}}_1 \circledast \underline{\mathbf{W}}$, where $\underline{\mathbf{K}}_0$ and $\underline{\mathbf{K}}_1$ are the trainable convolution weights. This is akin to the explorations of Rigamonti *et al.* [140], who demonstrated –in the context of learned sparse dictionaries– that complex filters/features can be expressed by linear combinations from a smaller number of separable filters to reduce computational complexity. Furthermore, this also has been exploited in the design of CNNs [141]. Finally, we highlight that in conventional CNNs, successive convolution layers cannot be assumed as separable filters, because they contain ReLUs between each convolution. Finally, the filters of the decoding path are defined by

$$\tilde{\underline{\mathbf{F}}}^{\intercal} = \begin{pmatrix} \tilde{\underline{\mathbf{F}}}_{\mathrm{LL}} \\ \tilde{\underline{\mathbf{F}}}_{\mathrm{LH}} \\ \tilde{\underline{\mathbf{F}}}_{\mathrm{HL}} \\ \tilde{\underline{\mathbf{F}}}_{\mathrm{HH}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{w}}_{\mathrm{LL}} \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}_0^{\intercal} \\ \tilde{\mathbf{w}}_{\mathrm{LH}} \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}_0^{\intercal} \\ \tilde{\mathbf{w}}_{\mathrm{HL}} \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}_0^{\intercal} \\ \tilde{\mathbf{w}}_{\mathrm{HH}} \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}_0^{\intercal} \end{pmatrix}. \tag{5.4}$$

The remainder of this subsection focuses on analyzing the characteristics of the encoding-decoding path in a process akin to Section 3.5. In this process, diverse CNNs are analyzed under ideal conditions, i.e., $\underline{\mathbf{K}} \circledast \underline{\mathbf{K}}^{\intercal} = \mathbf{I} \cdot c$. It can be anticipated that in practice, the filter $\underline{\mathbf{K}}$ inside the LWFSN is trained for a specific task (in our case noise reduction), which will likely result in a frame that is not tight, but which instead has desirable properties for the noise reduction task. Still, the ideal assumption is useful to test if the topology of the LWFSN allows for perfect reconstruction, at least under ideal conditions. To characterize the encoding-decoding path of the LWFSN we analyze the linear part of network $L(\cdot)$ in a process akin to Section 3.5, in the case of shrinkage networks it is possible to assume that the threshold values are equal to zero ($\underline{t} = \underline{0}$). In this condition $\tau^{\mathrm{LET}}_{(\underline{0})}(\cdot) = \mathbf{I}$ and the LWFSN becomes linear. This transforms Eq (5.1) into

$$\begin{aligned} \mathscr{P}\{L\}(\mathbf{y}) = & f_{(2\uparrow)}\left(f_{(2\downarrow)}\left(\mathbf{y} \circledast \underline{\mathbf{K}} \circledast \underline{\mathbf{W}}_{\mathrm{H}}\right)\right) \circledast \tilde{\underline{\mathbf{W}}}_{\mathrm{H}}^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}^{\intercal} + \\ & f_{(2\uparrow)}\left(f_{(2\downarrow)}\left(\mathbf{y} \circledast \underline{\mathbf{K}} \circledast \underline{\mathbf{W}}_{\mathrm{L}}\right)\right) \circledast \tilde{\underline{\mathbf{W}}}_{\mathrm{L}}^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}^{\intercal}. \end{aligned} \tag{5.5}$$

This expression can be further simplified to

$$\mathscr{P}\{L\}(\mathbf{y}) = f_{(2\uparrow)}\left(f_{(2\downarrow)}\left(\mathbf{y} \circledast \underline{\mathbf{K}} \circledast \underline{\mathbf{W}}\right)\right) \circledast \tilde{\underline{\mathbf{W}}}^{\intercal} \circledast \tilde{\underline{\mathbf{K}}}^{\intercal}. \tag{5.6}$$

For the DWT it holds that $\mathbf{p} = f_{(2\uparrow)}(f_{(2\downarrow)}(\mathbf{p}) \circledast \underline{\mathbf{W}}) \circledast \tilde{\underline{\mathbf{W}}}^{\intercal}$, therefore $\mathscr{P}\{L\}(\mathbf{y})$ is reduced to

$$\mathscr{P}\{L\}(\mathbf{y}) = \mathbf{y} \circledast \underline{\mathbf{K}} \circledast \tilde{\underline{\mathbf{K}}}^{\intercal}. \tag{5.7}$$

This result is the further simplified to

$$\mathscr{P}\{L\}(\mathbf{y}) = \mathbf{y} \cdot c. \tag{5.8}$$

If $c = 1$, then the LWFSN ensures perfect signal reconstruction. This proves that in the ideal scenario in which $\underline{\mathbf{K}}$ is a tight frame, the encoding-decoding path of the LWFSN ensures perfect reconstruction for any signal.

B Shrinkage path

In this chapter, we apply noise reduction on the high-frequency bands generated by a learned frame $\underline{\mathbf{F}}$ via a semi-hard thresholding function, which is based on the linear expansion of thresholds (LET) proposed by Blu and Luisier [108] and which is defined in Section 3.3.3-D in Eq. (3.33), which for self containment we show again here

$$\tau^{\text{LET}}_{(\underline{t})}(\mathbf{o}) = \sum_{n=0}^{N_T-1} a_n \cdot \tau_{(t_n)}(\mathbf{o}),$$

where a_n is the weighting factor assigned to each threshold, and in which all weighting factors should add up to unity and N_T is the total number of thresholds used. As thresholding function, we define $\tau(\cdot)$ as a semi-hard thresholding operation, based on the difference of Gaussians of Eq. (3.32), given by

$$\tau_{(t)}(d) = d \cdot \left(1 - \tanh(5 \cdot t) \cdot \exp\left(-\frac{d^4}{t^4}\right)\right). \tag{5.9}$$

where d is an element of the tensor of the detail bands $\underline{\mathbf{D}}$. Note that Eq. (3.32) has been augmented with the term term tanh$(\cdot)$, which approximates the sign function. This modification is motivated to allow $\tau_{(\cdot)}(\cdot)$ to also perform signal boost (t<0) in addition to thresholding (t>0) and the unity operation (t=0). The possibility of performing signal boost may be relevant in other inverse problems, such as image deblurring.

Finally, in order to justify the use of the semi-hard instead of soft-thresholding, we revisit two observations performed by Luisier [142]. First, the soft-thresholding function sets the image coefficients that fall below the threshold level to zero, despite the fact that natural images contain small coefficients which convey important texture information. In contrast, the semi-hard threshold used does not fully eliminate small components. Second, the soft-thresholding function is a biased estimator, while the semi-hard function is asymptotically unbiased for large image coefficients, which are most likely parts of the signal.

C Residual LWFSN

In contrast with conventional ReLU-based CNNs [8], [50], the LWFSN cannot be directly employed to produce the noise estimates required for residual noise reduction for two reasons. First, the thresholding functions applied to the high-pass framelet bands are designed to *eliminate* the noise, not to preserve it. Second, the absence of non-linearities in the low-frequency framelet band does not allow to cancel that part of the signal. Consequently, if it is desired to use the LWFSN in residual configuration, it must be modified as shown in Fig. 5.3. We refer to this modified architecture as *residual LWFSN* (rLWFSN). It should be noted that besides the main residual connection, there are two additional changes. First, the the semi-hard threshold $\tau^{\text{LET}}_{(\underline{t})}(\cdot)$ is replaced by a semi-hard clipping $\mathcal{C}^{\text{LET}}_{(\underline{t})}(d)$, which is defined by

$$\mathcal{C}^{\text{LET}}_{(\underline{t})}(d) = \hat{\boldsymbol{\eta}}_d = d - \tau^{\text{LET}}_{(t)}. \tag{5.10}$$

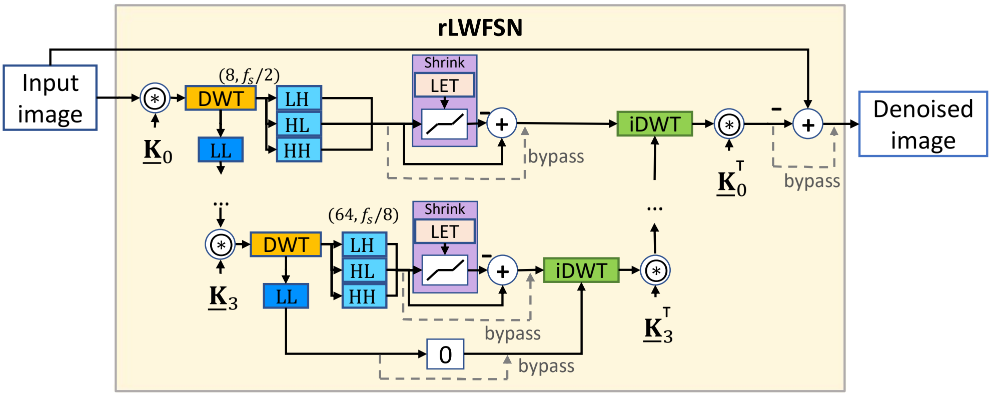


Figure 5.3. Modified LWFSN for residual noise reduction. In this configuration the noise is estimated at every scale so that it can be removed from the noisy input image with with the global residual connection. In the figure, the dashed lines denote the signal path if the network is bypassed to obtain the impulse response. Note that if the dotted bypasses are used, the bypassed network becomes equivalent to the bypassed non-residual LWFSN, thereby creating the same behavior.

The second alteration necessary to adapt the LWFSN for residual learning, is that the low-frequency band is set to zero at the deepest level, since in residual configuration the signal is suppressed. Note that this is akin to the low-frequency nulling used by Kwon and Ye [143]. Finally, the noise is reconstructed by the decoding path and subtracted from the noisy observation.

To conclude this section, the attention of the reader is directed to the diagram portrayed in Fig. 5.3, which shows that to compute the impulse response of the rLWFSN, it is sufficient to ignore the residual blocks and the suppression of the low-frequency band. This makes the signal path equivalent to the signal path of the conventional LWFSN. Hence, the same impulse response analysis also applies for this CNN.

5.3 Experiments and results

This section presents the experiments that compare the performance and properties of the proposed (r)LWFSN models with other state-of-the art models. The layout of this section is as follows. First, Section 5.3.1 explains the implementation details of the presented LWFSN model. Afterwards, Section 5.3.2 introduces the dataset, reference methods and metrics. Finally, Section 5.3.3 presents the training procedure and shows the results of the noise reduction experiments with the (residual) LWFSN model.

It can be observed that the content for the present chapter is primarily sourced from the article *Noise reduction in CT using Learned Wavelet-Frame Shrinkage Networks* [58]. In this chapter some applications are presented in the original text that have been excluded from this chapter. The reason for this omission is to improve the storyline of the thesis and the integration with the other chapters of this text. The excluded experiments are as follows. (1) The analysis of a CNN model which shows the conditions necessary for the TF U-Net to converge to a model with shrinkage functions. (2) Ablation studies of the proposed architecture highlighting the impact of the components of the LWFSN, as well as the effect of sharing the encoding and decoding filters in the reconstruction characteristics of the individual layers. In addition, during the realization of this thesis it was found that in the article that sourced this chapter a scaling error occurred in the computation of the HaarPSI metric. This error causes that there are a few differences between the results presented here and in the original article. A description of the error and the corrections that have been applied are presented in Appendix A.

5.3.1 Implementation of the LWFSN

The architecture of the implemented LWFSN system is depicted in Fig. 5.2 (a). The input image is first decomposed by the convolution tensors $\underline{\mathbf{K}}_{l,j}$, where l denotes the decomposition level and j the tensor index. The following step is a recursive dyadic decomposition implemented with the DWT, which enables multi-resolution processing and sets the high-frequency path, which can be filtered efficiently with wavelet-thresholding. At each level, we perform wavelet shrinkage with the linear expansion of thresholds (LET), where the threshold level and weight for every threshold unit is learned. The right part of the diagram represents the reconstruction.

Fig. 5.2 (b) is a more detailed description of the LET operation, where we use only two trainable vectors $\underline{t}$ and $\underline{b}$ at every decomposition level. The weights a_n and the thresholds t_n are computed, which are used in the final computation of the LET, and the dimensions and number of parameters per layer of the LWFSN are displayed in Table 5.1. The thresholds are initialized with uniform random values ranging between zero and 0.1, while vector $\underline{b}$ is initialized with unity values. All the CNNs in this paper are implemented in Pytorch [144] and the implementation of all the networks shown here is available in GitHub and on IEEE's Code Ocean [1].

5.3.2 Dataset, reference methods and metrics

A Dataset

For the experiments presented in this section, we use the full- and simulated reduced-dose brain CT scans (FDCT and QDCT, respectively) of the cancer imaging archive [145], [146]. The dataset contains pairs of full-dose CT scans and realistically simulated low-dose images. For testing, we apply 25 scans (892 slices), all of which have at least one lesion specified in the clinical information of the dataset. The remaining 25 scans are split into the training and validation sets with 12 subjects each (with 417 and 427 slices, respectively). Scan N293 was excluded from training/validation because the neck and shoulders were scanned as well. It was found that the slices of

[1] The code that replicates the most important results of this model is available at `https://github.com/LuisAlbertZM/demo_LWFSN_TMI` and interactive demo available at IEEE's code ocean `https://codeocean.com/capsule/9027829/tree/v1`

Level	Tensor	Dimensions per axis	Parameters count
0	$\underline{\mathbf{K}}_{0,0}$	$(1,4,2,2)$	16
0	$\underline{\mathbf{K}}_{0,1}$	$(4,8,2,2)$	128
0	$\underline{t}_0$	(5)	5
0	$\underline{b}_0$	(5)	5
1	$\underline{\mathbf{K}}_{1,0}$	$(8,8,2,2)$	256
1	$\underline{\mathbf{K}}_{1,1}$	$(8,16,2,2)$	512
1	$\underline{t}_1$	(5)	5
1	$\underline{b}_1$	(5)	5
2	$\underline{\mathbf{K}}_{2,0}$	$(16,16,2,2)$	1,024
2	$\underline{\mathbf{K}}_{2,1}$	$(16,32,2,2)$	2,048
2	$\underline{t}_2$	(5)	5
2	$\underline{b}_2$	(5)	5
3	$\underline{\mathbf{K}}_{3,0}$	$(32,32,2,2)$	4,096
3	$\underline{\mathbf{K}}_{3,1}$	$(32,64,2,2)$	8,192
3	$\underline{t}_3$	(5)	5
3	$\underline{b}_3$	(5)	5

Table 5.1. Tensor dimensions and parameters of the LWFSN encoder.

these regions contain photon starvation artifacts which are not representative of the rest of the scans. All the scans were acquired with helical geometry, slice thickness of 0.5 mm, pixel spacing/size of 0.48 mm and the slice size is 512 rows by 512 columns.

B Reference methods

For evaluating the noise reduction performance of the LWFSN, we have compared it against the TF U-Net and FBPConvNet. These are two state-of-the-art networks which have been successfully employed for artifact reduction in CT imaging. For fair comparison, all the CNNs use four decomposition levels. Furthermore, in the case of the TF U-Net the number of feature maps is decreased from 64 to 48 feature maps after the first convolution layer, since the large amount of feature rendered insufficient available hardware for the experiments (RTX-2080Ti GPU). For reference, we display the number of parameters of the methods under comparison in Table 5.2. Notice that the LWFSN only uses approximately 0.19% of the parameters of the described four-level TF U-Net and 0.45% of the parameters of FBPConvNet. In addition, Table 5.2 also displays the average forward propagation time for each of the CNNs, computed by averaging 500 forward propagations of an image with 512×512 pixels. Note that the LWFSN is 3.04 and 7.58 times faster that FBPConvNet and TF U-Net, respectively.

CNN	Total parameters	Forward prop. time [ms]
LWFSN	32,584	7.22
TF U-Net	17,657,904	54.77
FBPConvNet	7,306,497	22.02

Table 5.2. Number of trainable parameters and forward propagation time for the LWFSN TF U-Net and FBPConvNet CNNs.

C Image quality metrics

In order to compare the similarity between the processed images and the ground truth, we use the HaarPSI perceptual metric. Note that this metric requires the processed QDCT and ground-truth FDCT images ($\mathbf{x}$ and $\mathbf{y}$, respectively) to be bounded within the range $[0, 255]$. Therefore, prior to the computing the HaarPSI metric, we perform a pre-processing step in which the intensity of the images is clipped between maximum and minimum intensity, i.e. Min=0 [HU] and Max=80 [HU]. Afterwards, the clipped images are scaled to the range $[0, 255]$. This pre-processing is represented by function $P(\cdot)$, which is given by

$$P(\mathbf{x}) = 255 \cdot \frac{C(\mathbf{x}) - \text{Min}}{\text{Max} - \text{Min}} \ . \tag{5.11}$$

Here, function $C(\mathbf{x})$ is the clipping operation, defined by

$$C(\mathbf{x}_{i,j}) = \begin{cases} \text{Max} & \text{if Max} < \mathrm{x_{i,j}}; \\ \mathbf{x}_{i,j} & \text{if Max} \leq \mathrm{x_{i,j}} \leq \text{Min}; \\ \text{Min} & \text{if } \mathbf{x} < \text{Min}. \end{cases} \tag{5.12}$$

The maximum and minimum values are chosen because they are similar to the intensity ranges used to display the brain parenchyma [147]–[149]. Summarizing, the process to compute the HaarPSI HP$(\cdot)$, which is specified by

$$\text{HP}(\mathbf{x}, \mathbf{y}) = \text{HPSI}(P(\mathbf{x}), P(\mathbf{y})), \tag{5.13}$$

where HPSI$(\cdot)$ represents is HaarPSI computer program provided by Reisenhofer [97].

In addition to the HaarPSI, additional quality metrics used in this chapter are mean structural similarity index [95] (MSSIM) between the FDCT and processed QDCT, as well as the PSNR, where we assign as maximum value Max $= 80$ [HU], in concordance with the chosen intensity range used for visualization.

5.3.3 Noise reduction experiments

In order to fully characterize the proposed LWFSN and rLWFSN, we conduct diverse experiments. Section 5.3.3-A compares the noise reduction performance of the conventional LWFSN, the TF U-Net and the FBPConvNet models. Section 5.3.3-B shows an experiment that highlights the impact of using the semi-hard versus soft-thresholding in the LET. Meanwhile, Section 5.3.3-C discusses the performance of the residual LWFSN.

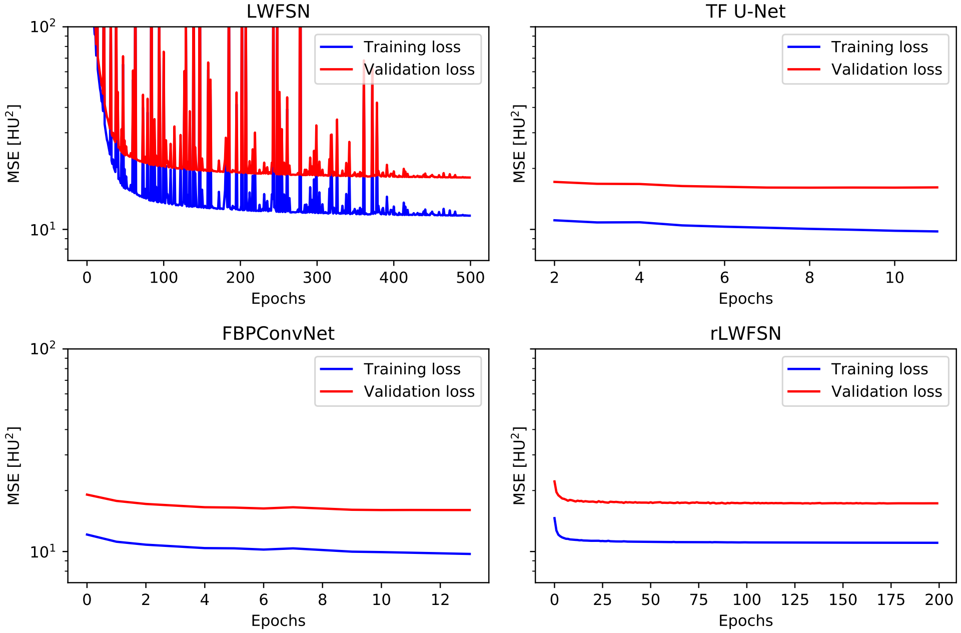


Figure 5.4. Training and validation loss curves for the main noise reduction architectures explored in this paper.

A Noise reduction performance for non-residual LWFSN

For the noise reduction experiments, we have trained the LWFSN, TF U-Net and FBPConvNet by minimizing the mean-squared error (MSE) loss function with Adam optimization, where the learning rate linearly decays from an initial value to zero. For the optimizer, the initial learning rate is empirically fine-tuned for each CNN, so that all of them achieve optimal performance. Specifically, the LWFSN is trained with an initial learning rate of 5×10^{-4} for 500 epochs. Meanwhile in the TF U-Net, the initial learning rate is set to 3×10^{-5} and the CNN is trained for 12 epochs. Finally, FBPConvNet is trained with an initial learning rate equal to 10^{-4} for 14 epochs. All the networks are trained directly in the HU scale. For training and testing, any pixels where zero-padding is applied within the CNNs are excluded. This is achieved by padding with reflections of the QDCT images prior to be processed and later crop the results back to the input dimensions. Fig. 5.4 showcases the training and validation losses for the tested CNNs, where it can be observed that the proposed LWFSN is slower in converging in terms of iterations. This is sensible because it is a non-residual CNN, so that it must learn to represent the image, which leads to larger errors in the initial iterations.

The noise reduction performance of the LWFSN, TF U-Net and FBPConvNet is summarized in Table 5.3, in which AU stands for *arbitrary units*. In Fig. 5.10, it is visible that the compared methods are preserving fine detail in the images, while reducing the noise considerably. For most patches, either the proposed LWFSN or FBPConvNet

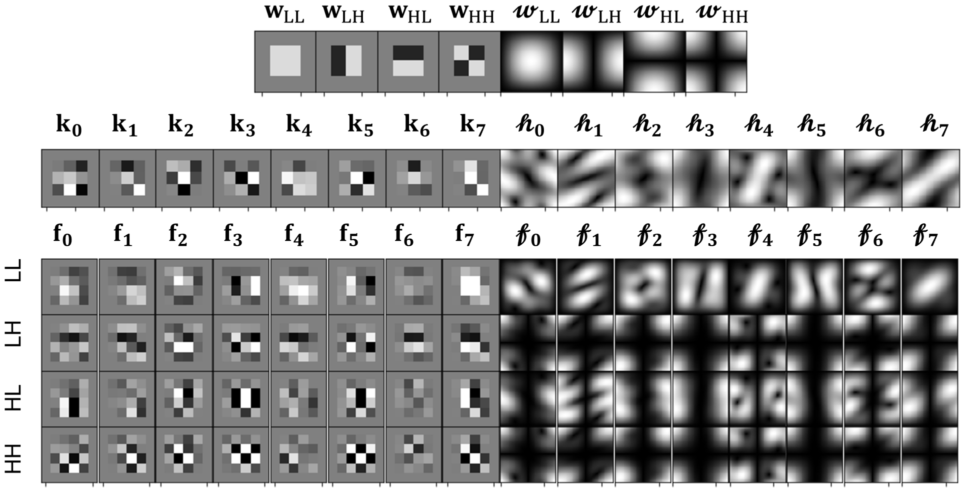

$\mathbf{w}_{\mathrm{LL}}$
$\mathbf{w}_{\mathrm{LH}}$
$\mathbf{w}_{\mathrm{HL}}$
$\mathbf{w}_{\mathrm{HH}}$
$\mathcal{w}_{\mathrm{LL}}$
$\mathcal{w}_{\mathrm{LH}}$
$\mathcal{w}_{\mathrm{HL}}$
$\mathcal{w}_{\mathrm{HH}}$
$\mathbf{k}_0$ $\mathbf{k}_1$ $\mathbf{k}_2$ $\mathbf{k}_3$ $\mathbf{k}_4$ $\mathbf{k}_5$ $\mathbf{k}_6$ $\mathbf{k}_7$
$\mathcal{k}_0$ $\mathcal{k}_1$ $\mathcal{k}_2$ $\mathcal{k}_3$ $\mathcal{k}_4$ $\mathcal{k}_5$ $\mathcal{k}_6$ $\mathcal{k}_7$
$\mathbf{f}_0$ $\mathbf{f}_1$ $\mathbf{f}_2$ $\mathbf{f}_3$ $\mathbf{f}_4$ $\mathbf{f}_5$ $\mathbf{f}_6$ $\mathbf{f}_7$
$\mathcal{f}_0$ $\mathcal{f}_1$ $\mathcal{f}_2$ $\mathcal{f}_3$ $\mathcal{f}_4$ $\mathcal{f}_5$ $\mathcal{f}_6$ $\mathcal{f}_7$
LL
LH
HL
HH

Figure 5.5. Wavelet frame learned by the LWFSN and its frequency localization characteristics, visualized for the first decomposition level. The first row depicts the elements of the convolution filter $\underline{\mathbf{W}}$ used in the DWT. The second row is the learned section of the frame $\underline{\mathbf{K}} = \underline{\mathbf{K}}_{0,0} \circledast \underline{\mathbf{K}}_{0,1}$. Finally, the third row of the figure depicts the equivalent frame $\underline{\mathbf{F}} = \underline{\mathbf{K}} \circledast \underline{\mathbf{W}}$. To the right of every tensor, the Fourier spectrum of the learned filters is shown (indicated by script letters).

perform best. When analyzing Table 5.3, it can be observed that FBPConvNet performs best in terms of the HaarPSI and PSNR when measuring over the entire dataset, while the best MSSIM is achieved by the TF U-Net. It should be noted that the performance gap between the compared CNNs is small.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
Baseline (QDCT)	0.796	0.654	25.65
LWFSN	0.836	0.670	27.19
TF U-Net	0.833	0.693	27.61
FBPConvNet	0.838	0.692	27.63

Table 5.3. HaarPSI, MSSIM and PSNR for noise reduction experiment.

Section 5.2.1-A proves that the LWFSN design ensures perfect signal reconstruction under the assumption that the encoding-decoding path is composed by tight frames. In this application, the LWFSN has been trained for noise reduction and there is no guarantee that the model has perfect reconstruction and/or that the filters of the encoding-decoding path are tight frames. In order to assess perfect reconstruction in the trained CNN, a unit impulse is supplied to the bypassed LWFSN. The result of this experiment is shown in Fig. 5.6, where it can be observed that the LWFSN reconstructs the unit pulse almost perfectly. Furthermore, the magnitude of the frequency response of the network's encoding-decoding path (i.e. the magnitude of the spectrum of

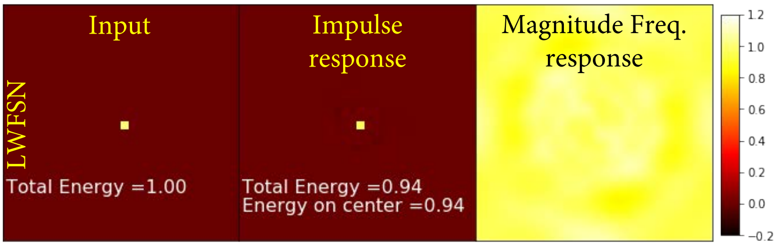


Figure 5.6. Impulse and frequency response of the encoding-decoding path of the LWFSN. In the left column, the unit impulse is displayed, that is supplied to the LWFSN. The center column illustrates the impulse response of the LWFSN. Finally, the right column portrays the Fourier spectrum of the impulse responses. The energy numbers in plots of the impulse response clearly indicate the dispersion of the energy in the samples surrounding the center pixel. It can be observed that reconstructed impulse is very close to the original input (a 2D delta Dirac function), which results in the nearly-flat unity spectrum at the right.

the impulse response) closely approximates an all-pass filter. This confirms that the encoding-decoding path of the LWFSN allows approximate perfect reconstruction, which is further corroborated by forward-propagating a QDCT slice through the bypassed LWFSN, as portrayed in Fig. 5.7. There it can be observed that the processed image is indistinguishable from the input.

The final characterization of the encoding-decoding path evaluates whether convolution filters at level l ($\underline{\mathbf{K}}_l$ and $\tilde{\underline{\mathbf{K}}}_l$) sufficiently approximate tight wavelet frames and their inverse. For this purpose, we perform the operation $\underline{\mathbf{K}}_l \circledast \tilde{\underline{\mathbf{K}}}_l^{\intercal}$. If $\underline{\mathbf{K}}_l$ is tight, it should hold that $\underline{\mathbf{K}}_l \circledast \tilde{\underline{\mathbf{K}}}_l^{\intercal} = \mathbf{I} \cdot c$. The result of this operation for the level $l = 1$ is shown in Fig. 5.8, where it becomes visually clear that $\underline{\mathbf{K}}_1 \circledast \tilde{\underline{\mathbf{K}}}_1^{\intercal} \neq \mathbf{I} \cdot c$. This means that the individual filters do not behave as tight frames. Despite the visual distortion in Fig. 5.8 with the filters at l=1, the LWFSN is still able to approximate perfect reconstruction as shown by the impulse response of the network shown in Fig. 5.6. Apparently, the other filters at higher levels are partly compensating for the imperfect reconstruction of the individual filters. It should be noted that if the filters in the encoding and decoding paths are shared i.e. $\tilde{\underline{\mathbf{K}}}_l = \underline{\mathbf{K}}_l$, the filter $\tilde{\underline{\mathbf{K}}}_l$ behaves more akin to a tight frame, but the noise reduction performance is lower [58].

Besides the noise reduction and reconstructed images, Fig. 5.5 presents the wavelet frame learned by the LWFSN on the first layer, which is computed by $\underline{\mathbf{F}}_0 = \underline{\mathbf{K}}_0 \circledast \underline{\mathbf{W}}$. Here, $\underline{\mathbf{K}}_0$ is the learned section of the wavelet frame $\underline{\mathbf{F}}_0$. For better understanding of this result, the convolution filter of the DWT $\underline{\mathbf{W}}$ is displayed in row form for convenience. In addition, the learned section $\underline{\mathbf{K}}_0$ and the resulting wavelet frame $\underline{\mathbf{F}}_0$ are presented as well.

Finally, Section 5.2.1-B mentions that the LWFSN model uses the LET [108] approach, which generates a composite threshold after a linear combination of simpler functions. Fig 5.9 displays all the learned thresholds for the first decomposition level, as well as the final combined function. This concludes the characterization of the proposed LWFSN architecture.

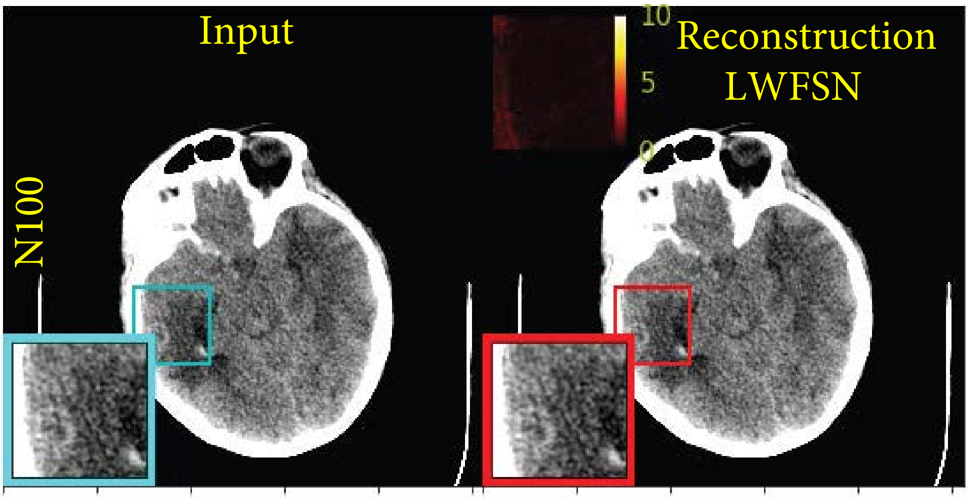


Figure 5.7. Example of input QDCT image (left column) and its reconstruction by the LWFSN with the shrinkage stage bypassed (right column). It can be noticed that the reconstructed image is very similar to the input. For better visualization, zoomed regions are shown (highlighted in red color for the processed images and in cyan for the noisy input).

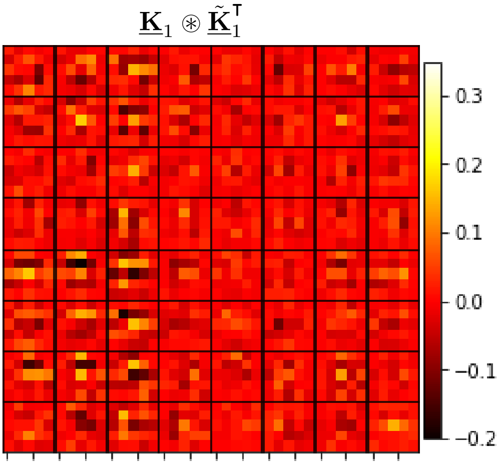


Figure 5.8. Convolution result $\underline{\mathbf{K}}_1 \circledast \tilde{\underline{\mathbf{K}}}_1^\intercal$ of the learned encoding $\underline{\mathbf{K}}_1$ and decoding filters $\tilde{\underline{\mathbf{K}}}_1$ at the first decomposition level. The specified color indicates the magnitude of the resulting signal. In the figure it can be observed that the filters $\underline{\mathbf{K}}_1$ and $\underline{\mathbf{K}}_1$ are not tight framelets because $\underline{\mathbf{K}}_1 \circledast \tilde{\underline{\mathbf{K}}}_1^\intercal \neq \mathbf{I}$, which is visible because there is a strong response outside the diagonal elements of the resulting tensor.

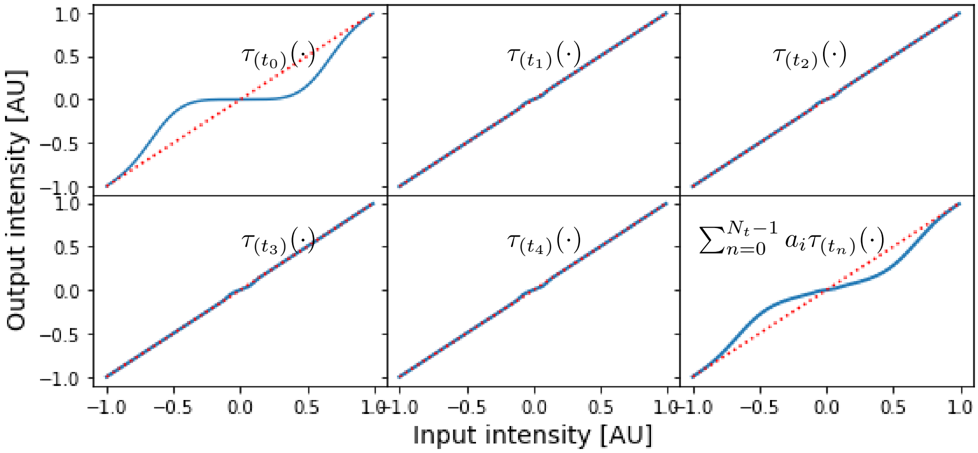


Figure 5.9. Learned thresholding functions of the shrinkage path of the LWFSN ($\mathcal{T}_{(t_0)}(\cdot)$, $\mathcal{T}_{(t_1)}(\cdot)$, $\mathcal{T}_{(t_2)}(\cdot)$, $\mathcal{T}_{(t_3)}(\cdot)$, $\mathcal{T}_{(t_4)}(\cdot)$) and combined LET shrinkage function.

B Semi-hard vs soft-thresholding in non-residual LWFSN

To study the influence of the semi-hard thresholding on the LET, this section shows an experiment, where the semi-hard thresholds of the LET are replaced by soft-thresholding functions. In the sequel, We refer to this variant as LWFSN soft, or LWFSN-S.

Given that the only difference between the LWFSN and LWFSN-S are the activation functions, the LWFSN-S is trained with the same parameters as the conventional LWFSN. The results of this experiment are shown in Table 5.4, from which it can be derived that the performance on the LWFSN-S is marginally lower than the semi-hard LWFSN. Since the performance gap between both designs is very small, three additional models of each architecture have been trained and the their performance metrics on the test set are averaged. The results of this experiment are displayed in Table 5.5 and confirm that the semi-hard thresholding used in the conventional LWFSN marginally outperforms soft-thresholding for this application.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
Baseline	0.796	0.654	25.65
LWFSN	0.836	0.670	27.19
LWFSN-S	0.833	0.667	26.99

Table 5.4. Noise reduction measurements comparing the LWFSN and LWFSN-S models, for three different performance indicators. Here, [AU] stands for arbitrary units. HaarPSI stands for the Haar perceptual similarity index.

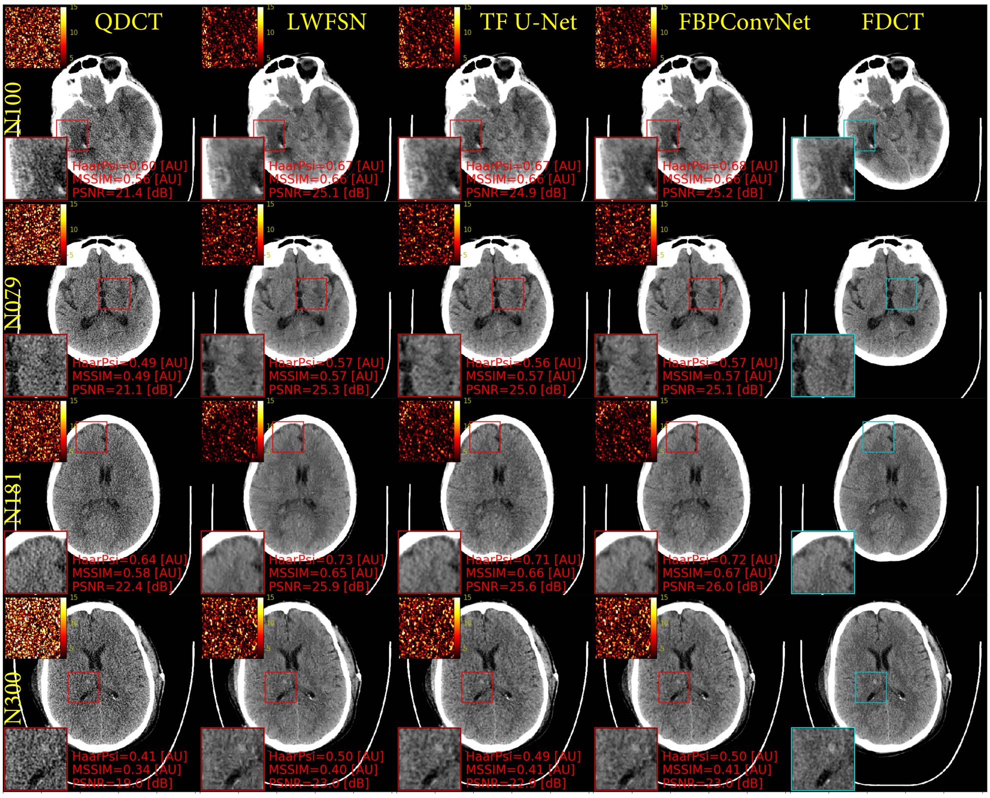


Figure 5.10. Input QDCT, processed QDCT and ground-truth FDCT images for diverse patients. Subject N100 contains a lesion caused by pulmonary carcinoma metastasis, N079 contains an acute stratal infarct, N181 has fluid accumulation and N300 a thalamic hemorrhage. The zoomed regions (highlighted in red color for the processed images and in cyan for the ground-truth) indicate the locations of the lesions. Furthermore, the difference between the zoomed regions and the ground-truth is also displayed. In addition, we also depict the HaarPSI, MISSIM and PSNR metrics on the displayed patches.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
LWFSN	0.835±4.4E-4	0.667±1.4E-3	27.12±0.032
LWFSN-S	0.832±1.1E-4	0.667±1.0E-3	26.98±0.027

Table 5.5. Average metrics for three LWFSN and LWFSN-S models.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
LWFSN	0.836	0.670	27.19
rLWFSN-NRW	0.836	0.672	27.53
rLWFSN	0.838	0.670	27.69

Table 5.6. HaarPSI, MSSIM and PSNR for the experiment that compares the non-residual LWFSN, the rLWFSN-NRW and the rLWFSN models.

C Residual vs non-residual LWFSN

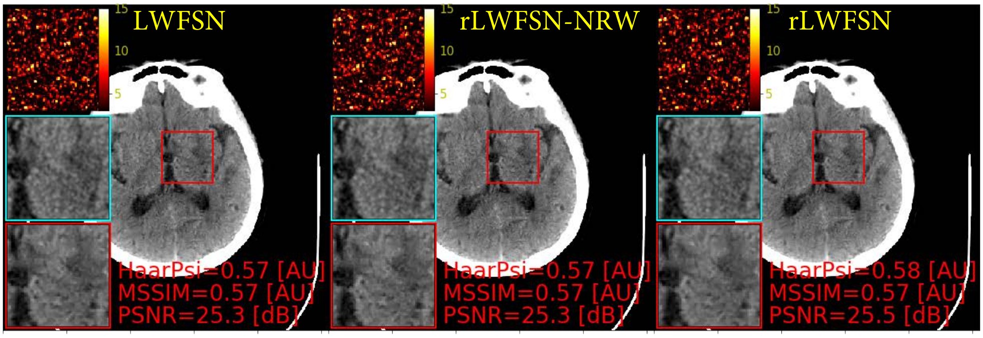


Figure 5.11. Slice of subject N079 processed by the LWFSN, the rLWFSN with the weights of the LWFSN (rLWFSN-NRW) and the rLWFSN. The metric results are indicated at the bottom of each figure.

In order to show the operation of the LWFSN in a residual configuration, the rLWFSN has been trained for noise reduction for 200 epochs with the Adam optimizer in which the initial learning rate is 3.5×10^{-4}. The training curve is shown in Fig. 5.4, where it can be seen that the convergence of the rLWFSN is faster and more stable than the conventional LWFSN. In addition, for reference, we have loaded also the parameters learned for the experiment in Section 5.3.3-A in the rLWFSN structure. This configuration is called residual LWFSN with non-residual weights, or rLWFSN-NRW. The purpose of showing this model is to demonstrate that the residual and non-residual are similar in how the signal is decomposed and reconstructed. This will be proven by loading the parameters that are learned in the non-residual model in the residual configuration. Meanwhile, the residual and non-residual models are different in the sense that the non-residual model shrinks low-amplitude values (typically associated to noise) in the decomposed signal, while preserving the high-amplitude

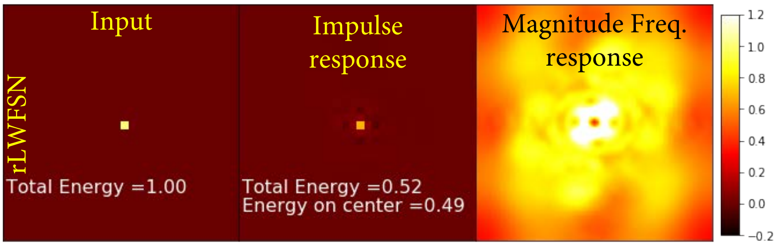


Figure 5.12. Impulse response of the rLWFSN. It should be noted that the image has a band-pass behavior and approximately half of the impulse response energy is spread out, resulting in frequency response variations.

values. On the contrary, residual models estimate the noise while suppressing the high-amplitude components of the decomposed signals. This is accounted for by replacing the shrinkage functions by clipping activations.

The performance metrics for this experiment are shown in Table 5.6. Furthermore, to complement this table, we display the processed slice by the LWFSN, rLWFSN-NRW and rLWFSN in Fig. 5.11. It can be observed that the rLWFSN performs better that the non-residual LWFSN, with a performance closer to FBPConvNet and the TF U-Net in terms of HaarPSI and MSSIM, and even better than the reference models in terms of PSNR. As expected, the rLWFSN-NRW performs very similar to the conventional LWFSN, which is a sensible result because both models share the same weights. Furthermore, it can be seen that the impulse response of the rLWFSN in Fig. 5.12 also approximates an impulse, but it has a band-pass behavior. While the origin of this behavior is not yet completely clear, we hypothesize that this may be caused by lowering the priority of signal reconstruction, since the low-frequency band is ignored in this configuration. In-depth research of this behavior is beyond the scope of this chapter and left for future work.

5.4 Discussion

The results of Section 5.3.3-A show that the presented models achieve noise reduction performance close to the state-of-the art CNNs, such as the TF U-Net and FBPConvNet. However, the proposed models have the benefit of a much lower computational cost and inference time. Furthermore, in the case of the LWFSN, it has been observed that its encoding-decoding path (globally) approximates perfect reconstruction without performing any regularization. This behavior is attributed to the topology of the network and to the fact that the input and target images only differ in the noise level. In the case of the rLWFSN, the perfect reconstruction is compromised, but better noise reduction performance is achieved as well as faster convergence and more stable training. This could be related to the fact that for residual models the signal does not need to be reconstructed (only the noise is estimated), which likely simplifies the optimization for this configuration.

The approximate perfect reconstruction shown by the encoding-decoding path of the LWFSN matches the theoretical analysis shown in Section 5.2.1-A, where ideal conditions are assumed, i.e. the encoding convolution frames are tight frames. Still, it has been found that the individual filters of the experimental LWFSN with independent encoding and decoding filters do not meet this condition. Nevertheless, the LWFSN approximates perfect reconstruction when all the systems interact. An important realization is that if the encoding and decoding filters are shared akin to BCD-Net [150] (i.e. $\tilde{\underline{\mathbf{K}}}_l = \underline{\mathbf{K}}_l$), the convolution filters display a behavior closer to tight frames. However, this resemblance comes with cost of lower noise reduction performance.

5.5 Conclusions

This chapter has proposed an encoding-decoding CNN that eliminates noise in reduced-dose CT acquisitions. The proposed CNN is called learned wavelet-frame shrinkage network (LWFSN) and the architecture is based on wavelet-based denoising algorithms such as BayesShrink [14] and the linear expansion of thresholds (LET) [108], which are well-known approaches for noise reduction in the wavelet domain.

Contributions A substantial advantage of the LWFSN over models such as TF U-Net and the FBPConvNet is that the design, has a low complexity, it is lightweight, fast and can be easily characterized. These advantages are elaborated in the following points.

1. *Well-interpretable operation*. The low complexity of the LWFSN and chosen non-linearity allows to prove theoretically and practically that the encoding-decoding path of the CNN can represent almost all of the spatial frequencies. The practical evidence about the signal reconstruction is achieved by measuring the impulse response of the encoding-decoding path of the trained LWFSN model. This test complements the theoretical analyses of Section 3.5. In addition, this result confirms that in the LWFSN model, the encoding-decoding path decomposes/reconstructs the input signal, while the shrinkage non-linearities eliminate parts of the signal decomposed by the encoder which are associated to noise.

   It should be noted that the proposed impulse response analysis is only possible because shrinkage non-linearities are chosen instead of the predominantly used ReLUs. The reason for this is that shrinkage functions guarantee than the model becomes linear when the threshold value is set to zero. It should be noted that measuring the impulse response is a widely accepted method to characterize linear systems. As a final remark, it can be observed that setting the bias/threshold level to zero in ReLU-based CNNs does not guarantee that the model becomes linear. Consequently, the analysis of the impulse response may not be a valid characterization of the behavior of those models and further studies are needed for that activation.

2. *Reduction of computational cost and number of parameters*. The proposed LWFSN model has a performance comparable to the TF U-Net and the FBPConvNet, while achieving a significant reduction in terms of the model size and execution time. Specifically, the LWFSN uses approximately 0.19% of the parameters of

the tested TF U-Net and 0.45% of the parameters of FBPConvNet. Furthermore, the LWFSN obtains inference 3.04 and 7.58-times faster than FBPConvNet and the TF U-Net, respectively.

The parameter and computation reduction is achieved with two main design choices. First, the model is very simple and uses less skip connections than the TF U-Net. This means that less feature maps are computed and stored. Second, the choice of a shrinkage activation allows for a reduction of feature maps, since ReLU-based models require more feature maps in order to avoid signal loss as discussed in Section 3.4.1.

Comparison with previous chapter. It can be observed that the training settings, network depth are different in Chapters 4 and 5. This does not allow for a fair comparison between the DHSN and the LWFSN models. In order to circumvent this limitation, Appendix B presents a comparison of the DHSN1, DHSN2, LWFSN and rLWFSN trained with the same settings as in this chapter. The experiments show that the rLWFSN, DHSN1 and DHSN2 achieve similar performance. Still, the rLWFSN uses less trainable parameters. The higher parameter count of the DHSN1 and DHSN2 owes to the fact that in these models, most of the trainable parameters are placed in between the encoder and decoder models (the OHWT-decomposed signal) in which the signal is high dimensional, which causes the convolution kernels to use more parameters and elevates the overall complexity of the models.

*Final conclusions.*The presented LWFSN model shows that the appropriate selection of design parameters based on basic understanding of the underlying signal concepts can lead to better interpretation of the system operation and insightful ways to address noise reduction. This is important to increase the trust in data-driven models that are used in medical systems.

It can be noted that the signal processing informed approach to design the CNN in this chapter is an alternative to the models presented in Chapter 4. Whereas in the previous chapter the design is based on a non-trainable framelet for decomposition and reconstruction of the signal, this chapter learns the signal decomposition and reconstruction as well as the denoising process. The advantage of the learnable transformation in this chapter is exploited for parameter reduction and simplifying the network. Still, the performance of both approaches is very similar and comparable as is the interpretability of the solutions. On the other hand, the representations in Chapter 4 are more accurate because the filters comply with perfect reconstruction and directionality *by design*, which may be attractive for specific applications.

As a final remark, we envision that this active approach to design CNNs may be useful in other pixel-level operations such as image segmentation, which is also widely applied in the medical domain.

The next chapter presents the final application of this text in which elements of the encoder-decoder CNNs shown in Chapters 4 and 5 are incorporated as part of a larger system. In fact, the solution developed in the next chapter is a hybrid system where elements of both approaches are combined to leverage their respective advantages. Specifically, Chapter 6 presents a model-based CNN which is used for denoising and material decomposition in a dual-energy cone-beam CT system.

CHAPTER 6

Regularized DE CBCT material decomposition with framelet-based CNN

6.1 Introduction

Chapters 4 and 5 have presented CNNs that have known properties such as ED paths with (almost) perfect reconstruction. The referred models are simpler and less computation-intensive than common CNNs, yet, the proposed models achieve performances comparable to state-of-the-art alternatives. Consequently, it is attractive to incorporate designs such as the models shown in Chapters 4 and 5 to other applications, such as model-based deep learning, in which CNNs are applied as a regularization technique within optimization-driven solutions to inverse problems. This chapter explores model-based material separation and noise reduction for dual-energy (DE) cone-beam (CB) CT. At the core of this application, there is a regularization CNN which incorporates elements of the models described in the previous chapters.

Background in dual-energy cone-beam CT. Denoising algorithms are important processing steps to improve the visualization of clinically relevant information in modern CT systems. In this chapter, noise reduction is applied to dual-energy (DE) cone-beam (CB) CT imaging, which is a novel technique and system concept where two spectral images are acquired simultaneously [151]. Each of the referred spectral images is tuned to absorb either most of the high-energy (HE) or the low-energy (LE) segments of the X-ray spectrum [151]. The additional observation of the energy-dependent X-ray attenuation process allows for better material quantification and/or virtual non-contrasted imaging. In order to exploit these capabilities, spectral decomposition algorithms are required. These algorithms work under the assumption that the sensed signals are composed by two energy-dependent sources of attenuation (e.g. two materials), which can be formulated as an equation system, of which the solution represents the material concentrations. With such an approach it should be noted that the inversion of noisy scans can boost the noise present in the material domain. Consequently, the inclusion of noise reduction stages as part of DE CBCT pipeline is critical to fully exploit the processing associated to this dual-energy system architecture.

Related work. Many techniques have been proposed for reducing noise in CT.

Some notable examples are total-variation denoising [13], sparse coding [31], framelets/wavelets [30] and more recently, convolutional neural networks (CNNs) [37], [50], [58]. In the case of dual-energy CT, specific denoising algorithms have been proposed. For example, the anti-correlated (AC) concept of Rudin, Osher and Fatemi (ROF) [152] and the iterative regularized image decomposition by Niu *et al.* [153] are proposed works that match with the denoising with dual-energy CT. Alternatively, CNNs such as the butterfly network (Butterfly-Net) of [154] have combined the DE CT material decomposition with noise reduction in a CNN that simultaneously estimates noise-free bone-water images from noisy dual-energy CT inputs.

Additional techniques which are applied to denoising single/dual-energy CT are *model-based* CNNs [121], which are designs incorporating physical modelling of the image acquisition process. Specifically, these models use iterative solutions to regularized least-squares problems as "templates", in which specific elements of the solution (e.g. regularization steps), are replaced by trainable components such as CNNs. This integration enhances the interpretability and data efficiency of the resulting model [155]. The main disadvantage of model-based CNNs is their sequential nature, which potentially increases the execution time. An example of a model-based CNN is FISTA-Net [156], which has been applied to remove artifacts in single-energy CT. Another example is the block-coordinate gradient descent (BCD) network [156], which simultaneously removes noise and estimates material concentrations in DE CT.

Challenges. Training CNNs for material decomposition and denoising of dual-energy CBCT introduces a number of specific challenges that are summarized as follows.

1. *Data characteristics*. DE CBCT data is noisy by nature because the photon flux that reaches the detector is divided into two sensing layers (i.e. low-energy and high-energy). This means the noise per layer is higher than conventional (single-layer) CBCT. Consequently, it is desirable to investigate the application of noise reduction algorithms for improving the visualization of information and/or to enhance the operation of downstream algorithms.
2. *Preservation of spectral information*. The noise reduction pipeline in DE CBCT should preserve the spectral information contained in the scan. This means that the noise reduction algorithm should not only improve the signal-to-noise ratio per channel, but also to preserve the intensity relationships between the low-energy and high-energy pictures.
3. *Data efficiency*. DE CBCT is an emerging technique where limited data are available. Consequently, it is desirable that the techniques applied to process these scans are data-efficient.

In order to meet the above-described requirements, it is attractive to use model-based CNNs such as FISTA-Net [111], since these models incorporate prior knowledge of the signal generation process. The FISTA-Net model is a CNN which is based on the original FISTA [125] approach. This technique is a solution for inverse problems that consists of alternating gradient descent updates and wavelet shrinkage-based regularization. It should be noted that FISTA is designed for fast convergence and one of its main features is the accelerated-gradient updates which decrease monotonically *by design*. Akin to FISTA, FISTA-Net employs also gradient descent, but FISTA-Net uses a more elaborate design in order to learn monotonically decreasing step-sizes. Furthermore, in FISTA-Net, the regularization CNN requires additional training to mimic the sparsity and perfect reconstruction of FISTA's wavelet-based regularization.

Proposed solution. This chapter explores material decomposition and denoising in DE CBCT. This application consists of mapping a noisy DE CBCT slice to their respective concentrations of bone and water materials. However, it can be observed that the DE CBCT slice contains noise. Consequently, the intended application should involve noise reduction steps besides the mapping requirement. For this purpose, this dissertation proposes an alternative method known as *regularized conjugate gradient network* (rCGN), which further leverages proximal gradient descent, framelet-based denoising and deep learning.

The proposed rCGN consists of two main steps. The first action employs a CNN model to reduce noise in the dual-energy domain, where the noise has lower intensity when compared to the mapped image with bone-water concentrations. The second step uses the dual-energy CT estimate generated in the first step as initialization to a conjugate gradient least-squares (CGLS) algorithm with proximal updates. In this second step, the CGLS method is applied to decompose DE CBCT scans into their bone-water components. At every iteration, an additional CNN is applied for regularization in the bone-water domain. The CGLS algorithm computes the optimal step-sizes for each DE CBCT slice being processed. It can be observed that the integration of the *adaptive* CGLS algorithm within the rCGN contrasts with alternative model-based deep learning-based solutions where the step-sizes are learned, and consequently, they become independent from the slice that is being processed. The latter concept with learned step-sizes limits the generalization of conventional models, since the optimization landscape of the model-based solution changes for different DE CBCT slices.

Rather than using conventional CNNs for regularization in the rCGN, this chapter uses a proximal CNN that leverages a tight framelet with directional filters and perfect reconstruction. In the framelet domain, shrinkage-based networks are applied to eliminate the noise. From previous text, it can be inferred that two of such models are applied within the rCGN, one in the dual-energy domain and another one in the bone-water domain.

Contributions and organization of this chapter. In summary, our contribution is to propose a new image-domain material decomposition and denoising CNN referred to as *regularized conjugate gradient network* (rCGN), which integrates framelets, proximal gradient descent and CNNs. The proposed network achieves excellent denosing performance and generalization, while being faster than FISTA-Net. It can be observed that the proposed rCGN mimics more closely proximal gradient models than alternative model-based CNNs, that is, the rCGN uses the well-known CGLS method at its core, while employing a framelet-based denoising CNN. This means that the operation is even more explainable than conventional model-based deep learning. The second benefit of this approach is the inherent flexibility and adaptivity because the concept can be optimized per slice and is based on generic optimization techniques.

The organization of this chapter is as follows. First, Section 6.2 introduces information on DE CBCT and material decomposition. Section 6.3 presents the proposed approach. Section 6.4 shows the experimental results, which are discussed in Section 6.5. Finally, Section 6.6 presents the concluding remarks of this chapter.

6.2 Background on proximal optimization and DE CBCT

Prior to addressing the proposed method, Section 6.2.1 provides key insights on proximal gradient methods for optimization and Section 6.2.2 discusses material decomposition of DE CBCT images.

6.2.1 Proximal gradient methods

Line-search optimization encompasses algorithms where an objective function $J(\cdot)$ (e.g. the mean squared error) is minimized by finding successive points $\underline{x}_n$, $\underline{x}_{n+1}, \ldots,$ $\underline{x}_N$, such that $J(\underline{x}_{n+1}){\leq}J(\underline{x}_n)$. Here, the minimization is achieved by moving from point $\underline{x}_n$ in the direction $\underline{d}_n$ by a step-size α_n [157]. This procedure is mathematically expressed by

$$\underline{x}_{n+1} = \underline{x}_n + \alpha_n \underline{d}_n. \tag{6.1}$$

A common example of line-search optimization algorithms is (steepest) gradient descent, where the direction $\underline{d}$ is given by the gradient of the objective function, i.e. $\underline{d}{=}\nabla J(\underline{x}_n)$. An alternative algorithm within this family is the *conjugate gradient least squares* (CGLS), which is used in this chapter. In contrast with SGD, in the CGLS method the current position $\underline{x}_n$ and its previous value $\underline{x}_{n-1}$ are used to compute the optimal direction $\underline{d}_n$ and step-size α_n. This concept allows for faster convergence than the simpler SGD.

The above paragraph summarizes gradient-based optimization, which requires that the objective function is convex and smooth. In more general cases, it may be needed to optimize functions that are still convex, but not necessarily smooth. For example, assume that $J(\cdot)$ has the form

$$J(\underline{x}) = F(\underline{x}) + G(\underline{x}). \tag{6.2}$$

Here, $F(\cdot)$ is smooth, differentiable and convex, while $G(\cdot)$ is convex as well, but not smooth (e.g. $\|\underline{x}\|_1$). In this case, it is to possible minimize this objective using *proximal gradient optimization*. In this technique, the above-described line search algorithm is employed to compute a new point $\underline{x}_{n+1}$ that minimizes $F(\cdot)$. At the point $\underline{x}_{n+1}$, a smaller optimization problem is set to minimize $G(\cdot)$. The solution of this problem is achieved by a *proximal operator* [158]. The proximal gradient solution is summarized by the expression

$$\underline{x}_{n+1} = \mathrm{prox}_{G(\underline{x}_n)}\big(\underline{x}_n + \alpha_n \underline{d}_n\big), \tag{6.3}$$

where Eq. (6.1) is augmented with the proximal operator $\mathrm{prox}_{G(\underline{x}_n)}(\cdot)$, which is the solution of the optimization problem $G(\cdot)$ at the point $\underline{z}_n = \underline{x}_n + \alpha_n \underline{d}_n$. A typical proximal operator is the soft-thresholding function, which is the solution for $G(\underline{x}_n) = \lambda\|\underline{x}_n\|_1$. An overview of proximal operators for specific objective functions can be found in the work by Wen *et al.* [159]. Finally, it should be noted that in model-based deep learning solutions, such as MoDL [160] and FISTA-Net [111], the proximal operator is replaced by a CNN.

6.2.2 Material decomposition in DE-CBCT

A. Sinogram-domain material decomposition

As mentioned earlier, spectral DE CBCT employs a detector that has two layers that are tuned to absorb photons in specific sections of the X-ray spectrum. Based on this configuration, the spectral X-ray attenuation of the object can be sufficiently approximated by a linear combination of two base materials, e.g. bone and water [161]. Consequently, the system is modelled by integrating the photon energy ε, which is exponentially attenuated and weighted by the sensing layers of the system, leading to

$$\mathbf{p}_l(\vec{r}) = \int_0^{\varepsilon_{\mathrm{M}}} S(\varepsilon) D_l(\varepsilon) \left(e^{-d_{\mathrm{B}} \cdot \mu_{\mathrm{B}}(\varepsilon) - d_{\mathrm{W}} \cdot \mu_{\mathrm{W}}(\varepsilon)} \right) \varepsilon \, d\varepsilon, \tag{6.4}$$

where $\mathbf{p}_l$ is a *sinogram*, which contains a set of measurements $\mathbf{p}_l(\vec{r})$, which reflect the measured intensity at a specific sensing layer l, with $l \in$[LE, HE]. Here, LE and HE stand for low-energy and high-energy, respectively, while ray $\vec{r}$ indicates the path followed by the X-ray from the source to an element in the detector. In addition, constant ε_{M} is the maximum energy emitted by the X-ray source. Furthermore, $S(\varepsilon)$ denotes the X-ray source spectrum, $D(\varepsilon)$ is the detector responsivity and function $D_l(\cdot)$ is the spectral responsivity of the layer l. Finally, the energy-dependent material attenuations are $\mu_{\mathrm{B}}(\cdot)$ and $\mu_{\mathrm{W}}(\cdot)$ for the bone and water materials, respectively. Here, variables d_{B} and d_{W} are the ray integrals of the volumes which indicate the concentration of each of the materials along a given path. The ray integrals are defined by

$$d_{\mathrm{B}} = \int \boldsymbol{\delta}_{\mathrm{B}}(\vec{r}) d\vec{r}; \quad d_{\mathrm{W}} = \int \boldsymbol{\delta}_{\mathrm{W}}(\vec{r}) d\vec{r}. \tag{6.5}$$

Here, variables $\boldsymbol{\delta}_{\mathrm{B}}$ and $\boldsymbol{\delta}_{\mathrm{W}}$ are material-concentration volumes. After estimating d_{B} and d_{W} for every ray $\vec{r}$, a CT image reconstruction technique is employed to obtain the values of $\boldsymbol{\delta}_{\mathrm{B}}$ and $\boldsymbol{\delta}_{\mathrm{W}}$. It should be noted a scanned object can be composed by many materials, therefore, the accuracy of the estimated material concentration may vary depending its composition.

B. Image-domain material decomposition

When the raw dual-energy sinogram is not accessible (e.g. the material estimation is applied as post-processing or dual-source or voltage-switching CBCT) scanning configurations, the material decomposition can be applied in the image domain under the assumption that the CT images produced by the low-energy and high-energy detectors are mono-energetic [154]. From this reference, this property is expressed by the mathematical model

$$\begin{aligned}\mu_{\mathrm{LE}} &= \mu_{\mathrm{B}}^{\mathrm{LE}}\delta_{\mathrm{B}} + \mu_{\mathrm{W}}^{\mathrm{LE}}\delta_{\mathrm{W}} + \eta_{\mathrm{LE}}\,, \\ \mu_{\mathrm{HE}} &= \mu_{\mathrm{B}}^{\mathrm{HE}}\delta_{\mathrm{B}} + \mu_{\mathrm{W}}^{\mathrm{HE}}\delta_{\mathrm{W}} + \eta_{\mathrm{HE}}\,.\end{aligned} \tag{6.6}$$

Here, μ_{LE} and μ_{HE} are individual voxels of the volumetric CT images $\boldsymbol{\mu}_{\mathrm{LE}}$ and $\boldsymbol{\mu}_{\mathrm{HE}}$, which are the CT reconstructed images for the low-energy and high-energy detector sinograms $\mathbf{p}_{\mathrm{LE}}$ and $\mathbf{p}_{\mathrm{HE}}$, respectively. Furthermore, in the equation, $\mu_{\mathrm{B}}^{\mathrm{LE}}$=$\mu_{\mathrm{B}}(\varepsilon_{\mathrm{LE}})$, is the bone attenuation for the energy level $\varepsilon_{\mathrm{LE}}$, which is the average sensed energy level for the low-energy detector. Consequently, the variables $\mu_{\mathrm{B}}^{\mathrm{LE}}$, $\mu_{\mathrm{B}}^{\mathrm{HE}}$, $\mu_{\mathrm{W}}^{\mathrm{LE}}$ and $\mu_{\mathrm{W}}^{\mathrm{HE}}$ are the bone and water images at the low-energy and high-energy levels, respectively. In addition, variables δ_{B} and δ_{W} are voxels of the volumetric images $\boldsymbol{\delta}_{\mathrm{B}}$ and $\boldsymbol{\delta}_{\mathrm{W}}$, which indicate the concentration of the bone and water, respectively. Expressing Eq. (6.6) in matrix form [1] results in the model

$$\begin{pmatrix}\mu_{\mathrm{HE}} & \mu_{\mathrm{LE}}\end{pmatrix} = \begin{pmatrix}\delta_{\mathrm{B}} & \delta_{\mathrm{W}}\end{pmatrix}\begin{pmatrix}\mu_{\mathrm{B}}^{\mathrm{LE}} & \mu_{\mathrm{B}}^{\mathrm{HE}} \\ \mu_{\mathrm{W}}^{\mathrm{LE}} & \mu_{\mathrm{W}}^{\mathrm{HE}}\end{pmatrix} + \begin{pmatrix}\eta_{\mathrm{LE}} & \eta_{\mathrm{HE}}\end{pmatrix}, \tag{6.7}$$

which is equivalent to

$$\underline{\mu} = \underline{\delta}\mathbf{a} + \underline{\eta}. \tag{6.8}$$

Here, vectors $\underline{\mu}$ and $\underline{\delta}$ represent the CT attenuation and material proportions for a given voxel, respectively. In addition, $\mathbf{a}$ is the matrix for the forward model, which contains the attenuations for the bone and water materials at the low-energy and high-energy bands. As a final remark for this section, it can be observed that assuming that μ_{LE} and μ_{HE} being mono-energetic is a simplification and renders this model less accurate than the sinogram-domain model of Section 6.2.2 A.

6.3 Methods

6.3.1 Overview of the proposed approach

As mentioned in Section 6.1, this chapter explores image-domain material decomposition and denoising for DE CBCT. Consequently, the starting point for this application is the image-domain model from Eq (6.8). It can be observed that the referred model considers only individual pixels. Extending this model to full DE CBCT slices , results in

$$\underline{\mathbf{D}} = \underline{\mathbf{M}} \circledast \underline{\mathbf{A}} + \underline{\mathbf{N}}\,. \tag{6.9}$$

Here, $\underline{\mathbf{D}}$ is a DE CBCT slice, whose entries are vectors $\underline{\mu}$ from Eq. (6.6), $\underline{\mathbf{M}}$ is a tensor containing the images whose materials are the bone and water concentrations δ_{B} and δ_{W}. Furthermore, tensor $\underline{\mathbf{N}}$ contains the noise components for the low-energy and high-energy images, respectively. Tensors $\underline{\mathbf{D}}$, $\underline{\mathbf{M}}$ and $\underline{\mathbf{N}}$ have dimensions $(1 \times 2 \times N_{\mathrm{h}} \times N_{\mathrm{v}})$, were N_{h} and N_{v} are the dimensions in the horizontal and vertical directions,

[1] In order to maintain consistency with the previous chapters, we have chosen to adopt row vectors instead of column vectors in this formulation.

respectively. Furthermore, tensor $\underline{\mathbf{A}}$ is the matrix $\mathbf{a}$ in Eq. (6.8), which is reshaped as a tensor with dimensions $(2 \times 2 \times 1 \times 1)$. This tensor models the relationship between the attenuations measured by DE CBCT and their corresponding bone-water concentrations.

In order to find the noiseless material image $\underline{\mathbf{M}}$ from Eq. (6.9), this thesis integrates the CGLS method with a data-driven regularizer that enforces smoothness. The resulting method is shown in Algorithm 1, where the CGLS part is the same method proposed by Kawata and Nacioglu [162] for algebraic reconstruction in CT. It should be noted that the referred algorithm has been conceived initially for a different application. However, it is a general solution for solving linear inverse problems, such as the application shown here. Consequently, the only adaptation that is required is to replace the forward CT model by matrix $\underline{\mathbf{A}}$. In addition, in this thesis, the gradient updates are regularized with the CNNs $\mathcal{P}_{\mathrm{BW}}(\cdot)$ and $\mathcal{P}_{\mathrm{DE}}(\cdot)$ that operate in the bone-water and dual-energy domains, respectively. The remainder of this section details the steps that compose the proposed solution. It can be observed that since this chapter is more application-oriented and builds upon applying the concepts of the previous chapters in a practical case, the method description starts readily with the algorithm and is explained in a straightforward manner.

Initialization. The first step in Algorithm 1, described in line 3 is the computation of the initial value of the material tensor $\underline{\mathbf{M}}_0$. This process consists of two sub-stages. The first sub-stage is a filtering procedure applied in the dual-energy domain, which is computed by the CNN $\mathcal{P}_{\mathrm{DE}}(\cdot)$. It should be noted that in this domain, the noise intensity is lower than in the bone-water domain, which should help to provide a good initialization to the CGLS part of the algorithm. It can be observed that the second input of $\mathcal{P}_{\mathrm{DE}}(\cdot)$ is an estimate of the noise level in the dual-energy domain, denoted by $\hat{\sigma}_{\mathrm{DE}}$. The second sub-stage (also in line 3) is to convert the initial noiseless dual-energy CBCT estimate to the bone-water domain. This is achieved by convolving the output of the regularization stage by the tensor $\underline{\mathbf{B}}$, whose elements are the inverse of $\underline{\mathbf{A}}$.

SGD step. The initialization of the CGLS algorithm requires the computation of an initial SGD optimization step, whose direction and step-size is obtained by the function $\mathrm{SGD\_DIRSTEP}(\cdot)$, as shown in Line 5. Afterwards, in Line 10 the intermediate material update is regularized with $\mathcal{P}_{\mathrm{BW}}(\cdot)$. Note that the regularizer employs an estimate of the noise level of the image in the bone-water domain ($\hat{\sigma}_{\mathrm{BW}}$).

CGLS iterations. After the completion of the initialization of the CGLS optimization, the remainder of Algorithm 1 updates and regularizes the material estimates with the actual CGLS steps and directions, which are given by routine $\mathrm{CGLS\_DIRSTEP}(\cdot)$ in Line 9 and with the network $\mathcal{P}_{\mathrm{BW}}(\cdot)$ in Line 10.

Retrieval of dual-energy and material estimates. The last steps of Algorithm 1 are shown in Lines 12 and 13. Both of these lines represent the final bone-water noiseless estimates, as well as the noiseless estimate of the DE CBCT image, respectively. It can be observed that the estimate of the DE CBCT is obtained by applying the forward model to the bone-water estimate. As a final remark, it should be noted that the operator $\odot$ in Lines 23 and 28 represent the inner product operation and not an element-wise multiplication. The next section describes the design of the networks $\mathcal{P}_{\mathrm{DE}}(\cdot)$ and $\mathcal{P}_{\mathrm{BW}}(\cdot)$.

Algorithm 1 Regularized CGLS CNN for DE CBCT decomp. and denoising.

```
function RCGN ( $\underline{\mathbf{D}}$, $\hat{\sigma}_{\mathrm{BW}}$, $\hat{\sigma}_{\mathrm{DE}}$)
    // Initialization
    $\underline{\mathbf{M}}_0 \leftarrow \mathcal{P}_{\mathrm{DE}}(\underline{\mathbf{D}},\ \hat{\sigma}_{\mathrm{DE}}) \circledast \underline{\mathbf{B}}$
    // The first step is SGD
    $\alpha_0, \underline{\mathbf{R}}_0 \leftarrow$ SGD_DIRSTEP $(\underline{\mathbf{M}}_0, \underline{\mathbf{D}})$
    $\underline{\mathbf{M}}_1 \leftarrow \mathcal{P}_{\mathrm{BW}}(\underline{\mathbf{M}}_0 + \alpha_0\underline{\mathbf{R}}_0,\ \hat{\sigma}_{\mathrm{BW}})$
    // Computing the remaining iterations with CGLS
    for $n \in 1, ..., N_{\mathrm{i}} - 1$ do
        $\alpha_n, \underline{\mathbf{R}}_n \leftarrow$ CGLS_DIRSTEP$(\underline{\mathbf{M}}_n, \underline{\mathbf{D}},\ \underline{\mathbf{R}}_{n-1})$
        $\underline{\mathbf{M}}_{n+1} \leftarrow \mathcal{P}_{\mathrm{BW}}(\underline{\mathbf{M}}_n + \alpha_n\underline{\mathbf{R}}_n,\ \hat{\sigma}_{\mathrm{BW}})$
    // Return values
    $\hat{\underline{\mathbf{M}}} \leftarrow \underline{\mathbf{M}}_{N_i-1}$                          // bone-water slice
    $\hat{\underline{\mathbf{D}}} \leftarrow \underline{\mathbf{M}}_{N_i-1} \circledast \underline{\mathbf{A}}$                     // dual-energy slice
return $\hat{\underline{\mathbf{D}}}$, $\hat{\underline{\mathbf{M}}}$

function SGD_DIRSTEP ( $\underline{\mathbf{M}}_n$, $\underline{\mathbf{D}}$)
    // Gradient step
    $\underline{\mathbf{R}}_n \leftarrow (\underline{\mathbf{D}} - \underline{\mathbf{M}}_n \circledast \underline{\mathbf{A}}) \circledast \underline{\mathbf{A}}^{\intercal}$                  // direction
    $\alpha_n \leftarrow \|\underline{\mathbf{R}}_n\|_2^2 / \|\underline{\mathbf{R}}_{n-1} \circledast \underline{\mathbf{A}}\|_2^2$                 // step-size
return $\alpha_n$, $\underline{\mathbf{R}}_n$

function CGLS_DIRSTEP ( $\underline{\mathbf{M}}_n$, $\underline{\mathbf{D}}$, $\underline{\mathbf{R}}_{n-1}$)
    // Intermediate steps
    $\underline{\mathbf{Q}}_n \leftarrow (\underline{\mathbf{D}} - \underline{\mathbf{M}}_n \circledast \underline{\mathbf{A}}) \circledast \underline{\mathbf{A}}^{\intercal}$                  // intermediate residual
    $\underline{\mathbf{\Gamma}} \leftarrow (\underline{\mathbf{Q}}_n \circledast \underline{\mathbf{A}}) \odot (\underline{\mathbf{R}}_{n-1} \circledast \underline{\mathbf{A}})$
    $\beta_{n-1} \leftarrow \underline{\mathbf{\Gamma}} / \|\underline{\mathbf{R}}_{n-1} \circledast \underline{\mathbf{A}}\|_2^2$                   // Intermediate step-size

    // Computing step-size for update
    $\underline{\mathbf{R}}_n \leftarrow \underline{\mathbf{Q}}_n - \beta_{n-1}\underline{\mathbf{R}}_{n-1}$                       // direction
    $\alpha_n \leftarrow (\underline{\mathbf{R}}_n \odot \underline{\mathbf{Q}}_n) / \|\underline{\mathbf{R}}_n \circledast \underline{\mathbf{A}}\|_2^2$                // step-size
return $\alpha_n$, $\underline{\mathbf{R}}_n$
```

6.3.2 Data-driven proximal operator

A Design of the denoising network

The data-driven proximal operators $\mathcal{P}_{\mathrm{DE}}(\cdot)$ and $\mathcal{P}_{\mathrm{BW}}(\cdot)$ are composed by two main elements which are listed as follows. (1) The encoding-decoding path is composed by the *overcomplete Haar wavelet transform* (OHWT) [57][2], which has the filters displayed in Fig. (6.1). This transform is chosen because of its compact and directional filters.

[2] Note that the number of feature maps of the OHWT increases with respect to the number of decomposition levels. Consequently, deeper networks require tensors with more parameters. This leads to a trade-off between the number of parameters and the network depth.

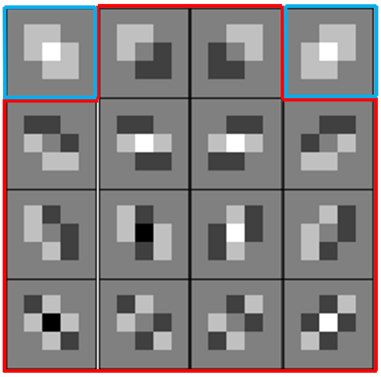


Figure 6.1. Graphical representation of the kernel of the overcomplete Haar wavelet transform (OHWT). The tensors highlighted in red have a high-pass nature and are referred to as $\underline{\mathbf{H}}_{\mathrm{H}}$, whereas the kernels highlighted in blue, represent the filters used to extract the low-frequency bands and are expressed by $\underline{\mathbf{H}}_{\mathrm{L}}$. The numerical values for the filters can be found in Chapter 4. As highlighted in the picture, this transform generates 14 high-pass and 2 low-pass bands, respectively. It can be observed that every input channel produces two low-pass channels/bands, which are further decomposed in deeper networks. Consequently, the number of channels expands with a growing number of decomposition levels.

(2) The second element of $\mathcal{P}(\cdot)$ is a group of noise reduction CNNs $\mathcal{D}(\cdot)$ which operate in the high-pass bands of the framelet-transformed signal. For a single-level network, the proposed regularization CNN $\mathcal{P}(\cdot)$ is given by

$$\mathcal{P}(\underline{\mathbf{M}}_i,\, \hat{\sigma}) = \mathcal{O}\Big(\underline{\mathbf{M}}_i \circledast \underline{\mathbf{H}}_{\mathrm{L}} \circledast \underline{\tilde{\mathbf{H}}}_{\mathrm{L}}^{\intercal} \circledast + \mathcal{D}_{(\hat{\sigma}_{\underline{\mathbf{M}}})}(\underline{\mathbf{M}}_i \circledast \underline{\mathbf{H}}_{\mathrm{H}}) \circledast \tilde{\mathbf{H}}_{\mathrm{H}}^{\intercal}\Big). \tag{6.10}$$

In this expression $\underline{\mathbf{M}}$ is the input signal, whereas $\hat{\sigma}_{\underline{\mathbf{M}}}$ is an estimate of its noise level. Moreover, tensors $\underline{\mathbf{H}}_{\mathrm{L}}$ and $\underline{\mathbf{H}}_{\mathrm{H}}$ are the low/high-pass filters of the forward OHWT, while $\underline{\tilde{\mathbf{H}}}_{\mathrm{L}}$ and $\underline{\tilde{\mathbf{H}}}_{\mathrm{H}}$ are their counterparts for the inverse transformation. Furthermore, the denoising network $\mathcal{D}_{(\cdot)}(\cdot)$ operates in the high-pass bands $\underline{\mathbf{X}} = \underline{\mathbf{M}}_i \circledast \underline{\mathbf{H}}_{\mathrm{H}}$ and it is defined by

$$\mathcal{D}_{(\hat{\sigma})}(\underline{\mathbf{X}}) = \underline{\mathbf{X}} - \mathcal{C}_{(\hat{\sigma}_{\underline{\mathbf{M}}} \cdot \underline{t}_1)}\big(\mathcal{C}_{(\hat{\sigma}_{\underline{\mathbf{M}}} \cdot \underline{t}_0)}(\underline{\mathbf{X}} \circledast \underline{\mathbf{K}}_0) \circledast \underline{\mathbf{K}}_1\big) \circledast \underline{\tilde{\mathbf{K}}}_1^{\intercal} \circledast \underline{\tilde{\mathbf{K}}}_0^{\intercal}. \tag{6.11}$$

Here, $\mathcal{C}_{(\cdot)}(\cdot)$ is the soft-clipping function [55]. In addition, $\underline{\mathbf{K}}_0$, $\underline{\mathbf{K}}_1$, $\underline{\tilde{\mathbf{K}}}_0$ and $\underline{\tilde{\mathbf{K}}}_1$ are convolution filters with spatial dimensions 1×1, which are learned during training. Furthermore, $\underline{t}_0$ and $\underline{t}_1$ are their respective biases/threshold levels. It should be noted that the denoising network $\mathcal{D}(\cdot)$ shown in Eq. (6.11) is executing a shrinkage function expressed in residual form [55]. Consequently, the proximal network $\mathcal{P}(\cdot)$ is simply executing a shrinkage operation in the framelet-decomposed signal, which is in line with well-established algorithms such as the original FISTA model [125].

The output layer of $\mathcal{P}(\cdot)$ depends on the domain where the network operates (i.e. bone-water or dual-energy), as further heuristics can be used to improve the operation of $\mathcal{P}(\cdot)$ on each specific domain. When $\mathcal{P}(\cdot)$ is applied to dual-energy images, the

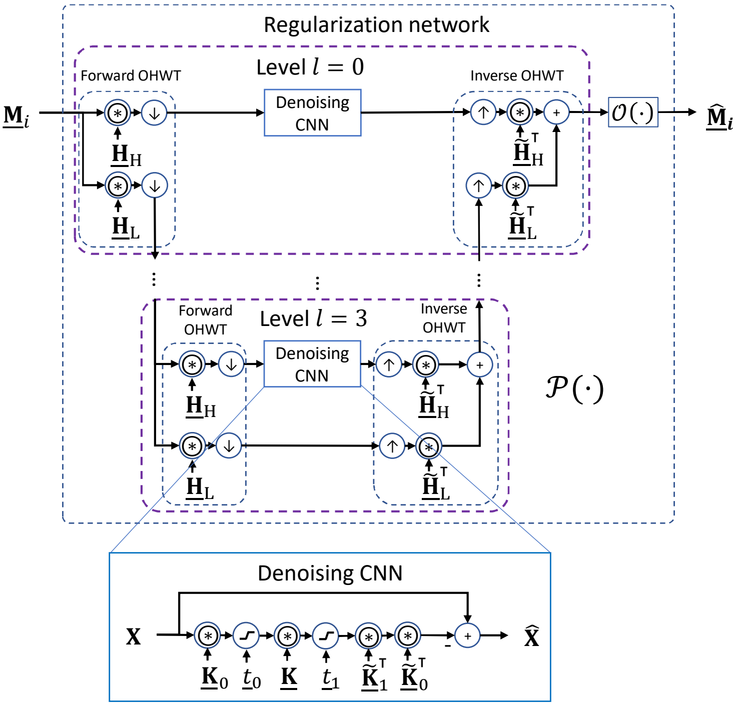

Regularization network
Level $l = 0$
Forward OHWT
Inverse OHWT
$\underline{\mathbf{M}}_i$
Denoising CNN
$\underline{\mathbf{H}}_{\mathrm{H}}$
$\underline{\mathbf{H}}_{\mathrm{L}}$
$\tilde{\underline{\mathbf{H}}}_{\mathrm{H}}^{\mathrm{T}}$
$\tilde{\underline{\mathbf{H}}}_{\mathrm{L}}^{\mathrm{T}}$
$\mathcal{O}(\cdot)$
$\hat{\underline{\mathbf{M}}}_i$
Level $l = 3$
Forward OHWT
Inverse OHWT
$\mathcal{P}(\cdot)$
Denoising CNN
$\mathbf{X}$
$\underline{\mathbf{K}}_0$
t_0
$\underline{\mathbf{K}}$
t_1
$\tilde{\underline{\mathbf{K}}}_1^{\mathrm{T}}$
$\tilde{\underline{\mathbf{K}}}_0^{\mathrm{T}}$
$\hat{\mathbf{X}}$

Figure 6.2. Proposed regularization CNN $\mathcal{P}(\cdot)$ with four decomposition levels. Here, tensors $\underline{\mathbf{H}}_{\mathrm{L}}$ and $\underline{\mathbf{H}}_{\mathrm{H}}$ are the convolution basis functions of the overcomplete Haar wavelet transform. Furthermore, $\tilde{\underline{\mathbf{H}}}_{\mathrm{L}}$ and $\tilde{\underline{\mathbf{H}}}_{\mathrm{H}}$ are their inverse transform counterparts.

output layer is the identity (i.e. $\mathcal{O}(\underline{\mathbf{M}}) = \underline{\mathbf{M}}$). Alternatively, when the model is applied to bone-water images, the output layer $\mathcal{O}(\cdot)$ is defined by

$$\mathcal{O}\left(\begin{pmatrix} m_{\mathrm{B}} \\ m_{\mathrm{W}} \end{pmatrix}\right) = \begin{pmatrix} (m_{\mathrm{B}})_+ - \frac{\mu_{\mathrm{W}}}{\mu_{\mathrm{B}}} \cdot (m_{\mathrm{W}})_- \\ (m_{\mathrm{W}})_+ - \frac{\mu_{\mathrm{B}}}{\mu_{\mathrm{W}}} \cdot (m_{\mathrm{B}})_- \end{pmatrix}. \tag{6.12}$$

Here, $(m_{\mathrm{B}}\,, m_{\mathrm{W}})$ are voxels of the intermediate estimates of the bone-water images $\underline{\mathbf{M}}_i$. It can be observed that this layer suppresses negative values in the bone and water concentration images, because negative material concentrations do not have physical meanings and clearly are an underestimation. We hypothesize that the negative values likely belong to the other material channel. Consequently, negative bone values are set to zero, and the suppressed part is scaled, negated and added to the water channel. In the case of negative water concentrations an analogous procedure is followed.

B. Estimation of noise intensity

The empirical estimates of the standard deviation in the bone-water and dual-energy images ($\hat{\sigma}_{\mathrm{BW}}$ and $\hat{\sigma}_{\mathrm{DE}}$, respectively) in Algorithm 1 are defined by

$$\hat{\sigma}_{\mathrm{BW}} = \sqrt{\hat{\sigma}_{\mathrm{B}}^2 + \hat{\sigma}_{\mathrm{W}}^2}; \quad \hat{\sigma}_{\mathrm{DE}} = \sqrt{\hat{\sigma}_{\mathrm{LE}}^2 + \hat{\sigma}_{\mathrm{HE}}^2}\,. \tag{6.13}$$

Here, $\hat{\sigma}_l$ is the noise standard deviation for layer/channel l∈[B,W,LE,HE], being computed with the median absolute deviation $\mathcal{M}(\cdot)$ of the high-pass voxels that belong to the head of the subject. This operation is mathematically described by

$$\hat{\sigma}_l = \mathcal{M}(\underline{\mathbf{Z}}_l) \cdot k, \tag{6.14}$$

in which $k = 1.4826$ is a constant value, $\underline{\mathbf{Z}}_l$ is a layer of the input image $\underline{\mathbf{Z}}$ and $\hat{\sigma}_l$ is its estimated noise standard deviation. Furthermore, operator $\mathcal{M}(\cdot)$ is defined by

$$\mathcal{M}(\underline{\mathbf{Z}}_l) = \begin{cases} \mathrm{msk\_med}(|\underline{\mathbf{Z}}_l * \mathbf{w}_{\mathrm{HH}}|, \mathbf{m}), & \text{for } \|\mathbf{m}\|_0 > 0; \\ \mathrm{median}(|\underline{\mathbf{Z}}_l * \mathbf{w}_{\mathrm{HH}}|), & \text{otherwise.} \end{cases} \tag{6.15}$$

Here, $\mathbf{w}_{\mathrm{HH}}$ is the diagonal filter of the 2D discrete wavelet transform with Haar basis function $\mathrm{msk\_med}(\cdot)$, which is the median computed over the non-zero elements of the mask $\mathbf{m}$, while function $\mathrm{median}(\cdot)$ is the conventional median, and denotes $\|\cdot\|_0$ is the L^0 norm (the number of non-zero entries). Furthermore, the mask $\mathbf{m}$ indicates the non-air pixels, which we arbitrarily define as the voxels with more than 500 Hounsfield units in the low-energy image. Consequently, $\mathbf{m}$ is expressed by

$$\mathbf{m} = \begin{cases} 1, & \text{if } 1000 \cdot \left(\frac{\mathbf{D}_{\mathrm{LE}} - \mu_{\mathrm{W}}(\varepsilon_{\mathrm{LE}})}{\mu_{\mathrm{W}}(\varepsilon_{\mathrm{LE}})} \right) > -500; \\ 0, & \text{otherwise.} \end{cases} \tag{6.16}$$

where $\mathbf{D}_{\mathrm{LE}}$ is the low-energy channel of the DE CBCT image being processed.

It should be noted that the current computation for the noise standard deviation estimate has two main limitations. (1) The computation in Eq. (6.13) inherently assumes that the noise in both layers is uncorrelated, which may not completely true. (2) The median absolute deviation estimator assumes that the noise is Gaussian, which is not true for CT imaging. Still, it is assumed that the noise level is reasonably constant for the head voxels within a given slice.

6.4 Experiments and results

6.4.1 Experimental setup

A. Dataset and reference methods

Synthetic dataset. For training, validation and testing, 19 pairs of noisy synthetic DE CBCT scans and their respective clean bone-water images have been employed. The synthetic scans use the head model by Simon *et al.* [163], which is based on the BrainWeb phantom [164]. The simulated acquisition geometry consists of a flat panel that has pixel size 0.742 mm, 396 rows and 512 columns. The detector responsitivity is taken from the paper of Engel *et al.* [165]. For reconstruction, the panel-to-source and isocenter-to-detector distances are set to 1.195 m and 0.39 m, respectively. With the described settings, 440 projections with angular spacing of $0.5°$ have been simulated. The reconstructions of the simulated projections have been performed with isotropic voxels

Filter name	Decomp. level	Kernel dimensions	Groups	Total parameters
$\mathbf{K}_0$	0	$(28 \times 56 \times 1 \times 1)$	1	1,568
$\mathbf{K}_1$	0	$(56 \times 112 \times 1 \times 1)$	2	3,136
$\underline{t}_0$	0	(56)	N.A.	56
$\underline{t}_1$	0	(112)	N.A.	112
$\mathbf{K}_0$	1	$(56 \times 112 \times 1 \times 1)$	2	3,136
$\mathbf{K}_1$	1	$(112 \times 224 \times 1 \times 1)$	4	6,272
$\underline{t}_0$	1	(112)	N.A.	112
$\underline{t}_1$	1	(224)	N.A.	224
$\mathbf{K}_0$	2	$(112 \times 224 \times 1 \times 1)$	4	6,272
$\mathbf{K}_1$	2	$(224 \times 448 \times 1 \times 1)$	8	12,544
$\underline{t}_0$	2	(224)	N.A.	224
$\underline{t}_1$	2	(448)	N.A.	448
$\mathbf{K}_0$	3	$(224 \times 448 \times 1 \times 1)$	8	12,544
$\mathbf{K}_1$	3	$(448 \times 896 \times 1 \times 1)$	16	25,088
$\underline{t}_0$	3	(448)	N.A.	448
$\underline{t}_1$	3	(896)	N.A.	896

Table 6.1. Kernel dimensions and number of parameters per layer of the denoising network within the rCGN algorithm.

CNN	Total parameters	Forward prop. time [ms]
rCGN	287,280	68.2
Butterfly-Net	301,954	15.6
FISTA-Net	75,176	102.5

Table 6.2. Number of trainable parameters and forward propagation time for the applied CNNs under testing.

with size of 1 mm. The low-energy and high-energy volumes have been reconstructed independently with the conjugate-gradient least-squares (CGLS) method [162], which has been applied for 40 iterations. The reconstruction algorithm has been implemented with the ASTRA toolbox [166]. After reconstruction, the volumes were inverted to its bone-water components to generate the ground-truth, while synthetically-generated noise is added to the reconstructed DE CBCT to serve as input to the networks.

The 19 synthetic scans are split in 5 scans for training, 5 for validation and 4 for testing. This generates a total of 990, 990 and 792 slices for training, validation and testing, respectively. For both synthetic and clinical scans, the energy levels used to compute $\underline{\mathbf{A}}$ have values of $\varepsilon_{\mathrm{LE}}$=60 keV and $\varepsilon_{\mathrm{HE}}$=75 keV.

Clinical scans. In addition to the synthetic scans, it is of primary importance to evaluate the performance of the methods under analysis when processing clinical scans. The scans selected for this section contain a contrast agent (iodine), which enhances the vascular structure of the brain. It should be noted that the models evaluated here assume that the CT slices are composed of bone and water and not iodine. In spite of this, it is hypothesized that most of the contrast agent will be displayed in the bone image because it is a dense material with a high attenuation coefficient for X-rays, which similar to the expected attenuation of bone. Just as in the case of the synthetic dataset, the energy levels are assumed to be $\varepsilon_{\mathrm{LE}} = 60$ keV and $\varepsilon_{\mathrm{HE}} = 75$ keV.

Reference methods. For comparison, this chapter uses the butterfly network (Butterfly-Net) [154] and FISTA-Net [111] as baseline solutions. Butterfly-Net is a CNN designed to denoise and convert DE CBCT scans to material concentrations. In our implementation of this model, the normalization layers are avoided, since it is observed that they introduce distortions to the estimated material images and affect the generalization of this model. This observation is in line with other imaging applications that avoid these layers as well [167], [168]. The second baseline for this chapter is FISTA-Net [111]. For this model, the forward matrix is set to the values of $\mathbf{a}$ as specified in Eq. (6.8). In addition, the number of input/output channels have been changed from one to two, so that it operates with dual-energy CBCT scans. Moreover, in line with the CT artifact-reduction application of the article by Xiang *et al.* [111], the Laplacian regularization matrix is not used for initialization. Finally, just as in the original article of FISTA-Net, 7 FISTA steps are employed.

Implementation of rCGN. The rCGN employed here has the following characteristics. First, the regularizers $\mathcal{P}_{\mathrm{DE}}(\cdot)$ and $\mathcal{P}_{\mathrm{BW}}(\cdot)$ are implemented with four decomposition levels, as shown in Fig. 6.2. Furthermore, the tensor dimensions and the grouping performed in the convolutions for parameter reduction are described in Table 6.1. Finally, it is worth mentioning that the trained rCGN model uses 10 CGLS iterations.

Model size and execution time. The total number of trainable parameters as well as the execution times for all models are displayed in Table 6.2. It can be noted that the smallest model in terms of parameters is FISTA-Net. However, it can be observed that it is the slowest model as well, because it is undecimated and consequently executes more operations. In contrast, Butterfly-Net is the model with the most parameters, but it is the fastest, which is caused by the non-iterative construction of this model, as opposed to rCGN and FISTA-Net. Finally, the rCGN model is the second smallest and fastest model.

A. Training procedure

For training the proposed network as well as Butterfly-Net, the L^1 loss is computed between the input noisy slices $\underline{\mathbf{D}}$ processed with a given CNN $G(\cdot)$ and the noise-free material-specific slice $\underline{\mathbf{M}}$. This is mathematically described by

$$\mathcal{L}_{\mathrm{dec}} = \|G(\underline{\mathbf{D}}) - \underline{\mathbf{M}}\|_1. \tag{6.17}$$

Here, $\mathcal{L}_{\mathrm{dec}}$ is the *material decomposition loss*. In the specific case of FISTA-Net, in addition to the similarity metric between the processed signal and the ground-truth, an additional loss is employed to ensure that the representation generated by the

Image type	CNN	PSNR [dB]	MSSIM
Low energy	Input	22.03	0.508
	rCGN	26.21	0.577
	Butterfly-Net	19.03	0.561
	FISTA-Net	26.84	0.683
High energy	Input	14.76	0.352
	rCGN	26.50	0.577
	Butterfly-Net	19.97	0.563
	FISTA-Net	27.22	0.681
Bone	Input	7.49	0.084
	rCGN	29.80	0.303
	Butterfly-Net	26.13	0.484
	FISTA-Net	33.44	0.561
Water	Input	-1.45	0.104
	rCGN	20.45	0.518
	Butterfly-Net	18.18	0.496
	FISTA-Net	23.62	0.581

Table 6.3. Obtained performance in terms of PSNR and MSSIM for noise reduction and material decomposition with DE CBCT images using different neural networks (rCGN, Butterfly-Net and FISTA-Net).

encoder of the regularization network of FISTA-Net is sparse ($\mathcal{L}_{\mathbf{spa}}$). Furthermore, FISTA-Net employs an additional loss term to train the encoding-decoding path of the regularizer to learn perfect signal reconstruction. This is referred to in the paper of Xiang *et al.* [111] as the symmetry loss ($\mathcal{L}_{\mathrm{sym}}$). Consequently, the complete loss for training FISTA-Net [111] is defined by

$$\mathcal{L}_{\mathrm{FISTAnet}} = \mathcal{L}_{\mathrm{dec}} + \lambda_{\mathrm{sym}}\mathcal{L}_{\mathrm{sym}} + \lambda_{\mathrm{spa}}\mathcal{L}_{\mathrm{spa}} \,, \tag{6.18}$$

where $\mathcal{L}_{\mathrm{FISTAnet}}$ is the loss term used to train FISTA-Net, while λ_{sym} and λ_{spa} define a trade-off for learning material decomposition, perfect reconstruction (sym) and sparsity.

All CNNs presented here are trained with Adam optimization with a linearly decaying learning rate. It can be observed that all presented models are very different in terms of their architecture, which means that the learning rate and the epochs necessary to train the models are different in each case. After an empirical evaluation, we have determined that the initial learning rate (iLR) for the proposed rCGN algorithm is set to $2{\times}10^{-3}$ and the model is trained for 200 iterations, while for Butterfly-Net, the iLR is $5{\times}10^{-4}$ and it is trained for 400 epochs. Finally, FISTA-Net is trained with an iLR of $5{\times}10^{-4}$ and trained for 400 iterations.

C. Image quality metrics

In order to compare the similarity between the processed images and the ground truth, the experiments presented here employ the *mean structural similarity index metric* (MSSIM) between the processed signal $\mathbf{X}$ and the ground truth $\mathbf{Y}$, as well as the peak signal-to-noise ratio ($\mathrm{PSNR}(\cdot)$), which is defined by

$$\mathrm{PSNR}(\mathbf{X},\mathbf{Y}) = 10 \cdot \log_{10}\left(\frac{(\mathrm{Max}(\mathbf{Y}))^2}{\mathrm{MSE}(\mathbf{X},\mathbf{Y})}\right). \tag{6.19}$$

Here, MSE denotes the mean-squared error, while $\mathrm{Max}(\mathbf{Y})$ is set to 55 [HU], which is the maximum value used for display in the CT images processed.

For display, the bone-water images have been scaled, so that their display intensities resemble Hounsfield units. For example, the water concentration $\mathbf{d}_{\mathrm{W}}$ should be centered around 0 units. For this purpose, the water image is scaled by

$$\mathbf{d}_{\mathrm{W}}^{\mathrm{PHU}} = (\mathbf{d}_{\mathrm{W}} - 1) \cdot 1000, \tag{6.20}$$

where $\mathbf{d}_{\mathrm{W}}^{\mathrm{PHU}}$ is the water image scaled in what is referred here as *pseudo Hounsfield units*. It should be noted that this scaling sets to zero when the concentration of water is 100% ($\mathbf{d}_{\mathrm{W}}$=1). Similarly, the bone image $\mathbf{d}_{\mathrm{B}}$ is scaled by

$$\mathbf{d}_{\mathrm{B}}^{\mathrm{PHU}} = \mathbf{d}_{\mathrm{B}} \cdot 1000, \tag{6.21}$$

where $\mathbf{d}_{\mathrm{B}}^{\mathrm{PHU}}$ is the image $\mathbf{d}_{\mathrm{B}}$ in pseudo Hounsfield units.

6.4.2 Experiments

Synthetic scans. Using the described experimental setup, we have trained the rCGN, Butterfly-Net and FISTA-Net. The performance measurements for the synthetic dataset can be found in Table 6.3. In addition, Fig. 6.3 displays the dual-energy CT slices of one of the synthetic scans, as well as the resulting bone-water and dual-energy CT estimates, produced by the tested CNNs. From Fig. 6.3, and Table 6.3, it can be observed that all the networks are able to significantly remove the noise present in the simulated noisy acquisitions. However, the estimates produced by FISTA-Net are the best in terms of the quality metrics, followed by the rCGN. It can be noticed that despite the differences in terms of quality metrics, for this dataset, the appearance of the estimates produced by the rCGN and FISTA-Net are virtually similar to each other with almost indistinguishable differences. This is not the case for the slice processed with Butterfly-Net, which has a significant amount of remaining low-frequency noise. This is likely caused by the network model, since the Butterfly-Net architecture is single-scale and relatively shallow. This means that its receptive field is small when compared to the rCGN and FISTA-Net.

Clinical scans. In addition to the synthetic dataset, the tested CNNs are also evaluated in the clinically-acquired CBCT slices, as displayed in Figs. 6.4, 6.5 and 6.6. The referred figures show the low-energy, water and bone images processed by the rCGN, Butterfly-Net and FISTA-Net, respectively. It should be noted that the high-energy

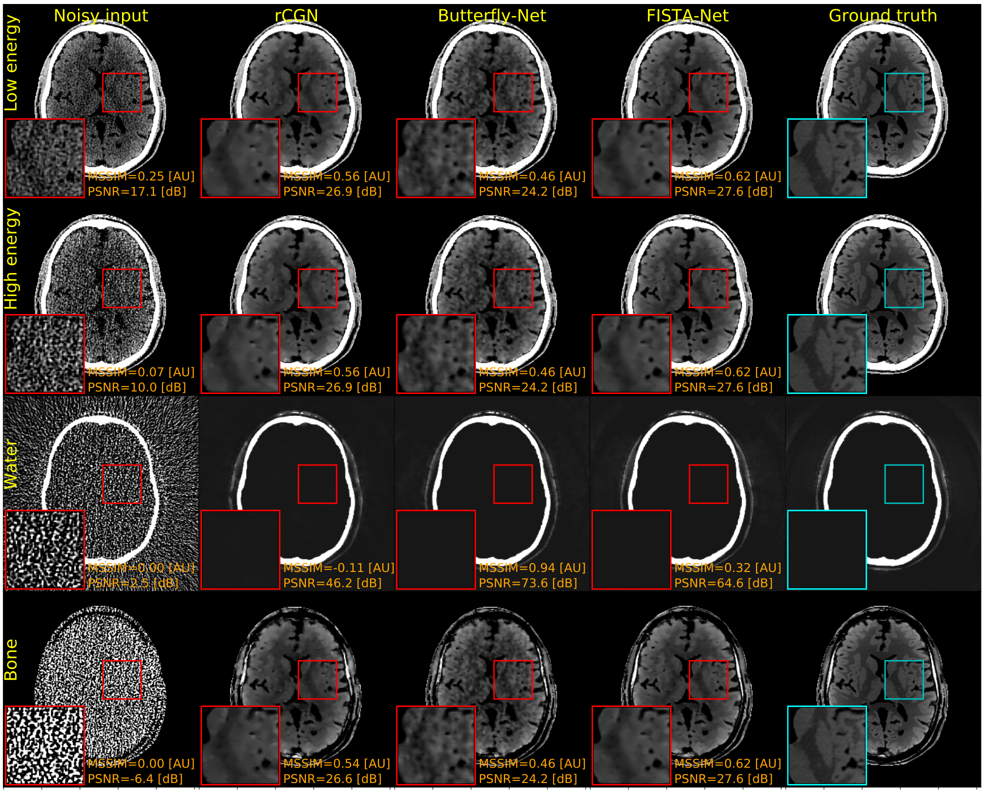


Figure 6.3. Dual-energy CT slices and their corresponding bone-water concentration images. The figure shows the noisy inputs, the ground-truth and the processed slices with the rCGN, Butterfly-Net and FISTA-Net. In the figure, the columns present the visual results of each of the networks being tested, while the rows are the image channels for the processed slice.

slices are excluded because the low-energy and high-energy images have a similar appearance.

To start the analysis in clinically-acquired images, it can be observed in Fig. 6.4 that all the networks improve the visibility of the lesions and produce high-quality low-energy images. In fact, there are no significant differences between the appearance of the low-energy images processed by the three networks.

When analyzing the water concentration components shown in Fig. 6.5, it can be observed that the most uniform images are produced by the rCGN. In this case, the second-best model is FISTA-Net, for which the image still shows some residual low-frequency noise. When observing the image produced by Butterfly-Net, it can be noted that the parenchyma is almost an exact copy of the low-energy image. This is specifically noticeable because most of the contrast agent is still visible in the images produced by this network. This result suggests that Butterfly-Net does not generalize well and it learns to copy the parenchyma from the low-energy to the water image, while suppressing the skull. An important realization from Fig. 6.5 is that the water-

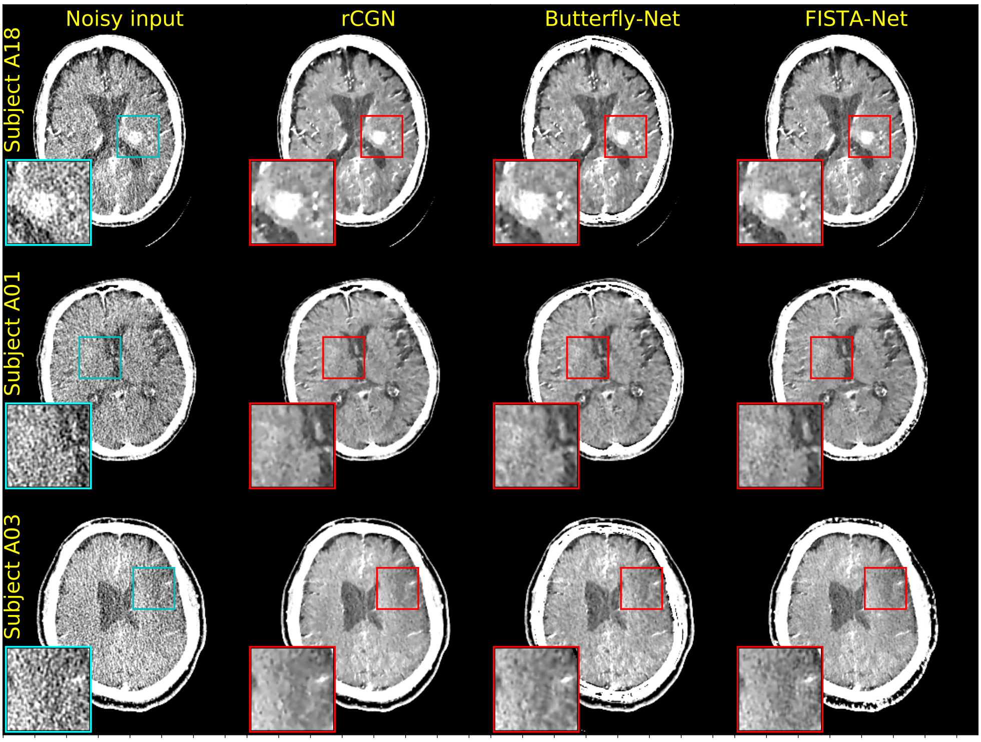


Figure 6.4. Low-energy image processed with the rCGN, butterfly network and FISTA-Net as well as the noisy input. For all the subjects lesions found in the scan are highlighted.

concentration images produced by the rCGN and FISTA-Net show beam-hardening in the vicinity of the skull. This is especially visible in Subjects A18 and A03 in Fig. 6.5. This behavior is expected, due to the modeling limitations of the image-domain material decomposition, which implicitly assumes that dual-energy images are mono-energetic. In can be noted that this is a simplification of the poly-energetic imaging process. This aspect can be circumvented by the use of a sinogram-domain decomposition, which is beyond the scope of this thesis.

The bone-concentration images displayed in Fig. 6.6 show that the proposed rCGN model produces smoother results with lower noise levels than FISTA-Net and Butterfly-Net. However, it should be noted that for the rCGN, the added smoothness comes at the cost of loosing some fine details (see subject A03). Finally, Fig. 6.7 shows the effect of the analyzed CNNs in the bone image, to improve the visualization of the contrast agent in the patients by displaying the maximum intensity projection in the bone concentration image over 55 slices. In Fig. 6.7, it can be observed that all the CNNs greatly improve the visualization of the vascular structure of the brain. In this case, the network that produces the clearest vessel structures is FISTA-Net, followed very closely by the rCGN. Here, Butterfly-Net seems to be darker than the rest of the methods, which indicates that this estimate is biased towards lower concentrations.

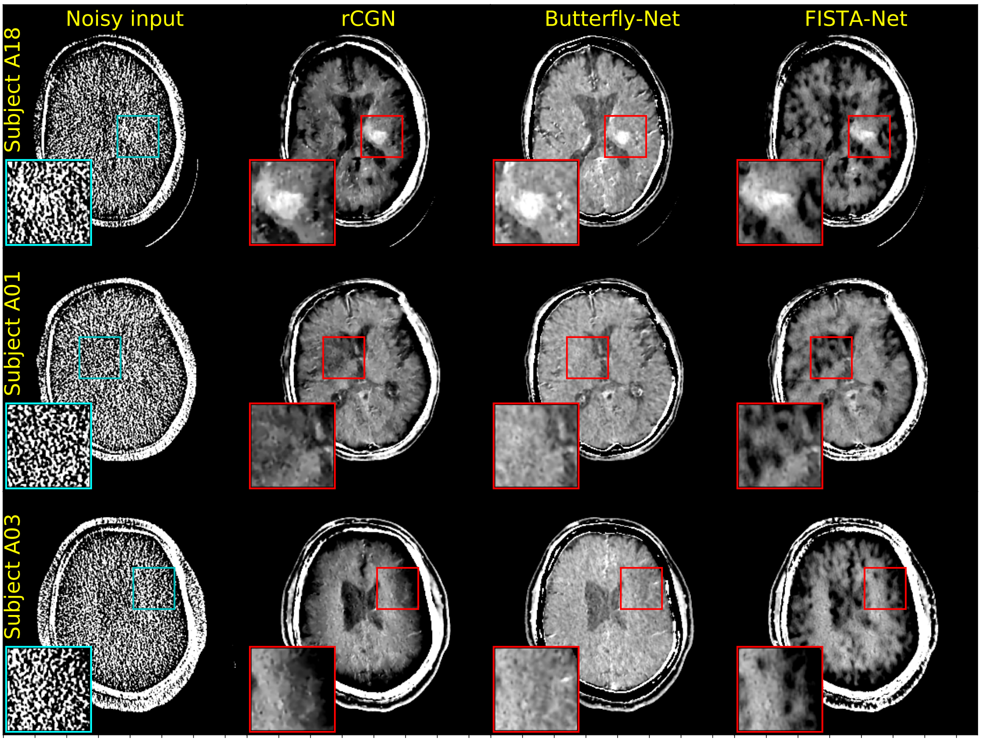


Figure 6.5. Corresponding water component to the scans of Fig. 6.4, which are processed with the rCGN, butterfly netwok and FISTA-Net as well as the noisy input. For all the subjects lesions found in the scan are highlighted.

6.5 Discussion and limitations

Discussion. The presented rCGN model is able to produce high-quality estimates which have performance metric results close to FISTA-Net for the synthetic scans. In addition, it can be observed in Fig. 6.3 that the images produced by the rCGN and FISTA-Net are almost indistinguishable from each other for this dataset, while the rCGN is considerably faster. Furthermore, it can be observed that the rCGN model also generalizes better to real clinical scans than FISTA-Net and Butterfly-Net. The improved generalization of the rCGN can be attributed to a few reasons which are as follows. (1) The proposed model leverages a sparse and directional transformation, which is already compliant with perfect reconstruction. (2) The noise reduction network architecture is very simple and therefore less likely to overfit on the synthetic dataset. (3) The proposed approach leverages prior knowledge about the strength of the noise, which allows to adapt the denoising process. (4) The CGLS is an adaptive optimization approach, which computes the optimal step size for every slice being processed. This contrasts with FISTA-Net, where the step sizes are learned. Henceforth, they are determined by the training procedure and not by the slice that is

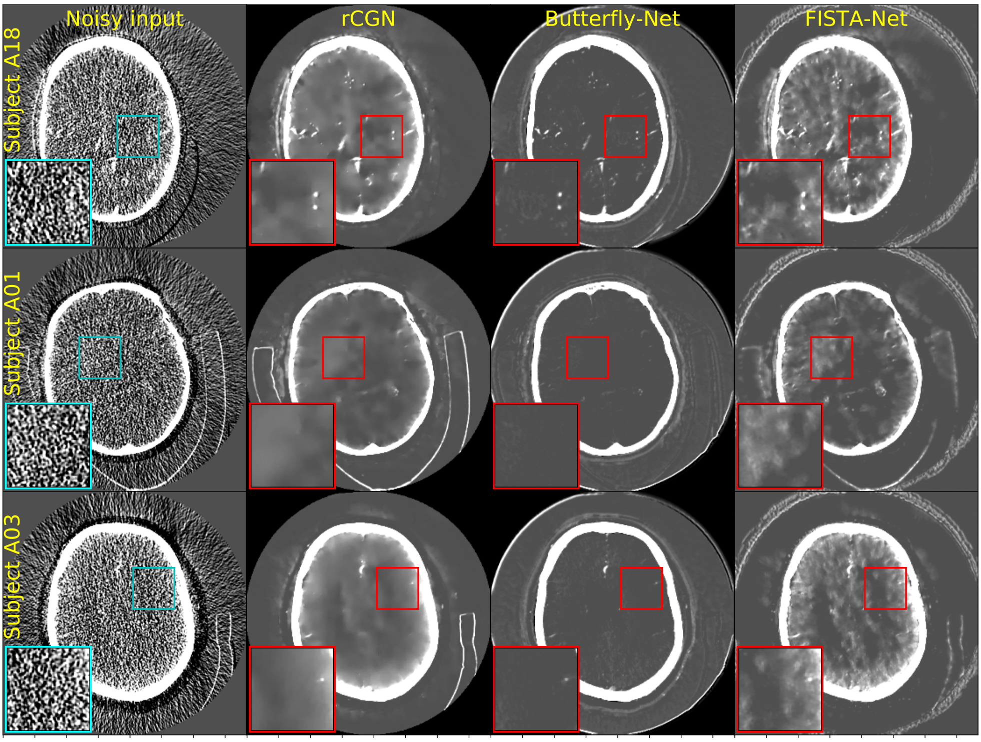


Figure 6.6. Corresponding bone component to the scans of Fig. 6.4, which are processed with the rCGN, Butterfly-Net and FISTA-Net as well as the noisy input. For all the subjects lesions found in the scan are highlighted.

being processed.

Regarding Butterfly-Net, two major issues can be observed. First, its receptive field is small, which causes that this model performs worse than the rCGN and FISTA-Net on the synthetic images. The second problem of Butterfly-Net is that it *learns* the forward/inverse model. This contrasts with FISTA-Net and the rCGN, which leverage the physics model of the attenuation process. This limitation causes that Butterfly-Net overfits on the synthetic scans and decreases its performance when evaluated on the real clinical images. This is especially visible in the water concentrations displayed in Fig. 6.5.

Limitations. The image-domain model presented in Eq. (6.6) is a simplification of Eq. (6.4), and neglects the wide spectrum of energy levels occurring in the dual-energy CBCT scan. This simplification causes beam hardening in the vicinity of the skull for some slices (e.g. in the water image of subject A03 shown in Fig. 6.5). Improvements to the current approach can be achieved by decomposing the signal in the sinogram domain, where it is possible to leverage a more accurate model of the imaging process (see Eq. (6.4)). It should be noted that this requires a more elaborate prepossessing of the data, using the sinogram and the acquisition geometry. This is beyond of the scope of this chapter and is considered for future work.

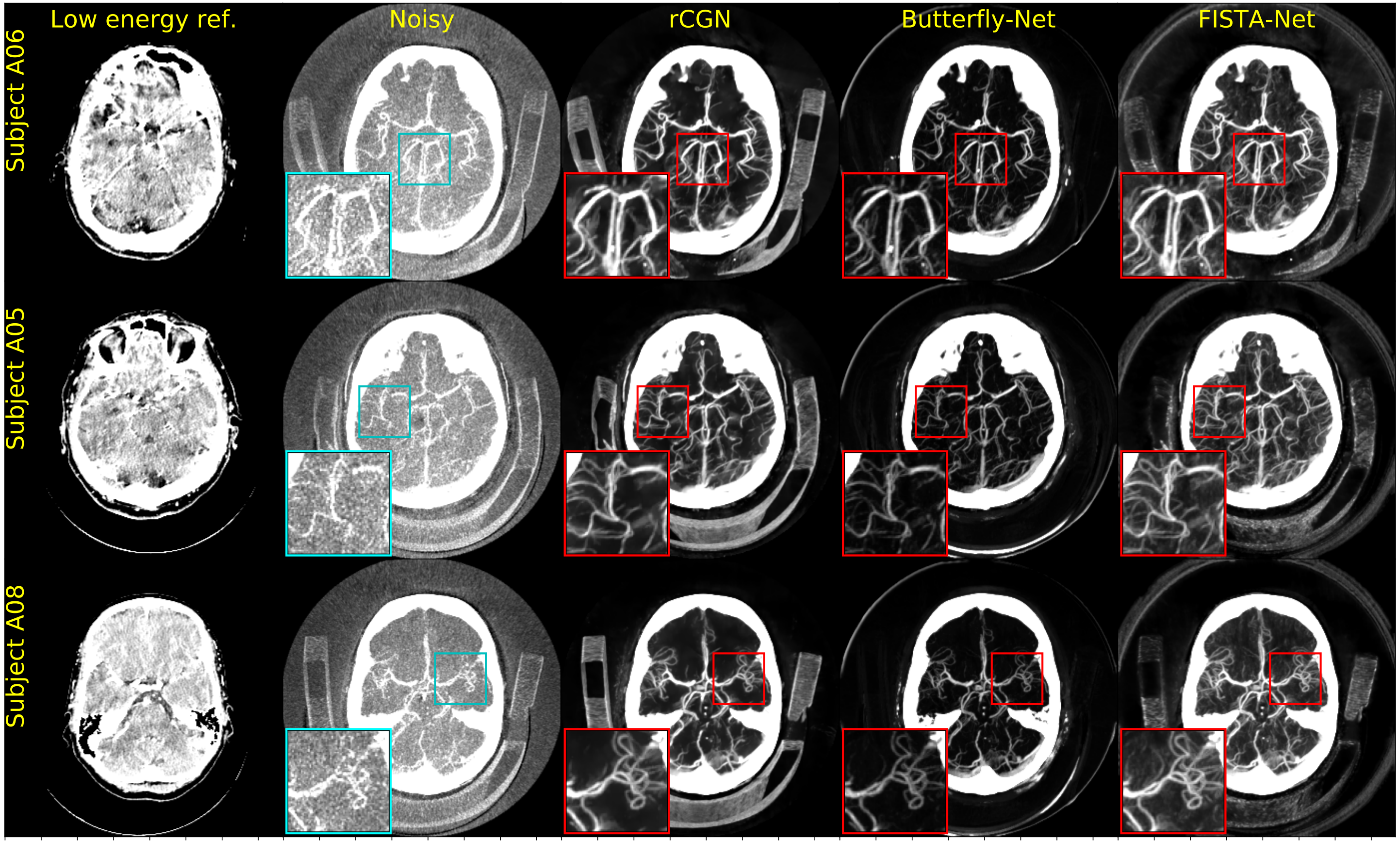


Figure 6.7. Bone concentration images with maximum-intensity projection. The presented images show how the application of the evaluated rCGN, Butterfly-Net and FISTA-Net models improve the contrast agent that is present in the vascular structure of the brain.

6.6 Conclusions

This chapter has presented a solution for noise reduction and material decomposition in dual-energy CBCT. In order to address this problem, a new model-based CNN known as *regularized conjugate gradient network* (rCGN) is proposed. The proposed model leverages the conjugate gradient least squares algorithm (CGLS), which is an iterative method for solving linear inverse problems. At every CGLS step, a proximal operator represented by a CNN enforces smoothness. The smooth-enforcing CNN is simple and leverages an existing framelet (the OHWT) that provides properties such as directionality, sparsity and perfect reconstruction. This design choice circumvents the use of explicit training to achieve such properties. The integration of these concepts, leads to a highly interpretable model-based CNN, since it mimics closely other proximal methods for solving inverse problems. The rCGN model has two main contributions which are listed below.

1. The rCGN avoids the more complex step-size scheduling of FISTA-Net by introducing the CGLS algorithm as optimization backbone. This means that in the rCGN the step-size is adaptively computed for every slice that is being processed.
2. The use of an existing framelet with perfect reconstruction and sparsity properties improves the interpretation of the model and it may be one of the reasons why this design is able to generalize better than the reference methods. In addition, the presented model has an adaptive behavior to the noise level, which is reminiscent of previously used wavelet-based denoising algorithms [14] and which has been exploited by the multi-scale sparse coding network [48].

The proposed rCGN model performs closely to FISTA-Net in material separation and denoising of dual-energy CBCT synthetic scans and generalizes better to clinical scans than the reference models, despite the fact that it is trained only with a limited amount of simulated scans. Furthermore, an additional advantage of the rCGN model is that it offers a faster execution than the FISTA-Net.

Model-based deep learning is an application where CNNs are a part of a larger system. As mentioned in this chapter, within model-based deep learning an optimization algorithm is employed to solve an inverse problem and part of this solution is a CNN that is employed for learned regularization. It can be observed that typically in spite of the more interpretable behavior of model-based deep learning, at its core a conventional "black box" CNN is employed. We have noted this aspect and considered it as an opportunity for employing some of the design elements of previous chapters to enhance the interpretability of these systems. For example, the rCGN employs the OHWT, which was first developed as the encoding-decoding path of the DHSN models from Chapter 5. Furthermore, the noise reduction network within the regularizer is inspired on the residual LWFSN model from Chapter 6, which proved to have high performance, while being compact. Finally, the idea of incorporating adaptivity to the denoising process comes from the multi-scale sparse coding network [48], which is first addressed in Chapter 3 as a means to improve the generalization of noise reduction models.

CHAPTER 7

Conclusions

This concluding chapter summarizes the main learning points of the conducted thesis research. The structure of this chapter is as follows. First, Section 7.1 summarizes the conclusions of each individual chapter. Afterwards, Section 7.2 addresses the research questions presented in Chapter 1 and discusses the contributions and answers to these questions. Finally, Section 7.3 discusses the future outlook for the research on further integration of signal processing and deep learning models.

7.1 Conclusions of the individual chapters

Chapter 2 introduces noise reduction techniques, such as total-variation denoising, wavelets, low-rank methods, as well as deep learning elements such as CNNs and autoencoders. These concepts form the foundations for the research of this thesis. In addition, this chapter introduces also the mathematical notation and relevant image quality metrics that are used in the experiments of the succeeding chapters.

Chapter 3 commences with the analysis of neural networks in terms of signal propagation and analysis. More specifically, existing signal processing concepts, as well as the theory of deep convolutional framelets [40] are developed for a solid theoretical description of ED CNNs. This approach strengthens the theory of deep convolutional framelets and extends it to more general networks with shrinkage and clipping activations. In addition, this chapter presents a framework that is used to describe and analyze CNNs to find potential limitations of these models. The found limitations on the analyzed CNNs motivate the design choices used in the following chapters.

Chapter 4 presents a new framelet transformation with compact and directional filters. The developed framelet is referred to as the overcomplete Haar wavelet transform (OHWT), which is a special case of the dual-tree complex wavelet transform. Afterwards, this new framelet is employed as encoding-decoding structure in two CNN models, referred to as the *dual-Haar shrinkage networks* 1 and 2 (addressed as *DHSN1* and *DHSN2*, respectively). Both of these networks are designed for removing noise in CT images and are inspired by the Neigh-shrink [54] approach for noise reduction. The followed technique consists of using a CNN to provide element-wise weights that suppress noisy components in the high-pass bands of the framelet decomposition, while preserving the detail information. The motivation for the integration of framelets and CNNs in the DHSN1 and DHNS2 models is to use trainable parts only to eliminate the noise from the signal and to handle the image decomposition/reconstruction with framelets that have known properties.

The DHSN1 and DHSN2 models achieve good noise reduction performance, which is comparable with other state-of-the-art models (the RED CNN and FBPConvNet) and produce image estimates that display excellent quality when visualizing patches of clinically-relevant structures. Furthermore, to demonstrate the advantages of using an ED path with known properties, this chapter shows also an unsupervised noise reduction pipeline, which leverages the sparsity properties of the proposed DHSN2. The results demonstrate that the tight integration of signal processing and deep learning leads to simpler models with more interpretable operation.

Chapter 5 proposes a CNN architecture for noise reduction, called *learned wavelet-frame shrinkage network*, or LWFSN, and its residual counterpart, the rLWFSN. In these networks, the encoding-decoding (ED) path is learned instead of a fixed structure as used previously. Furthermore, the ED path in this model is linear and performs the decomposition/reconstruction of the input signal, while the noise is suppressed by a shrinkage stage that is based on a linear expansion of thresholds [108].

The ED path of the LWFSN model is designed to achieve perfect reconstruction under ideal assumptions, as shown by the analysis in Section 5.2.1-A. Furthermore, the use of shrinkage non-linearities enables linear model behavior when the threshold is set to zero. This method is used to demonstrate that the linear part of the model operates in such way that the encoder and decoder decompose and reconstruct the signal in an almost perfect way, while the noise reduction behavior of the system is mainly caused by the shrinkage non-linearities. The described behavior is confirmed and validated empirically by using the impulse and frequency responses of the system. At the time of this research, it was the first time that such analysis was applied to a CNN.

Finally, it is worth mentioning that the LWFSN presented in Chapter 5 employs only a fraction of the training parameters of conventional CNNs (< 1%), while it offers a very short inference time (more than 7 times faster than the reference thigh-frame U-Net). However, the LWFSN performs similar to state-of-the-art alternatives, such as the tight frame (TF) U-Net and FBPConvNet for denoising low-dose CT images.

Chapter 6 Chapter 6 is more application-oriented and employs the concepts from preceding chapters, by incorporating a framelet-based CNN into a model that decomposes dual-energy (DE) cone-beam (CB) CT into material-specific images. The proposed model is referred to as *regularized conjugate gradient network* (rCGN) and incorporates framelet-based regularization, CNNs and conjugate gradient least squares (CGLS) optimization. It is observed that the proposed approach is closer to conventional iterative soft thresholding algorithms (ISTAs) than alternative model-based solutions such as FISTA-Net. This means that the proposed rCGN is more interpretable, since ISTA-like algoriths are designed with concrete assumptions about the signal-generation process. The rCGN and FISTA-Net models are trained with simulated DE CBCT scans and evaluated in both, synthetic and clinically-acquired images. The obtained results show that the proposed rCGN uses only 66% of the execution time of FISTA-Net, while achieving comparable peak signal-to-noise ratio and mean structural similarity index when evaluated with synthetic scans. However, the proposed rCGN generalizes better when evaluated with clinically-acquired scans.

7.2 Discussion on the research questions

RQ 1: Integrating conventional signal processing into CNNs

RQ1a. *What are the consequences of the integration of non-trainable framelets as ED path in noise reduction CNNs?*

Chapter 4 The integration of non-trainable framelets into the convolution structure of CNNs has two main consequences. On one hand, the use of non-trainable framelets as ED path in CNNs limits the tasks that the CNN learns, because the decomposition and signal reconstruction are already provided by the framelet transformation. On the other hand, the use of non-trainable filters decreases the flexibility of the model. This should be mitigated by choosing a suitable representation based on prior knowledge about the task and the nature of the data. Both of the addressed consequences are discussed in more detail below.

The main elements that should be considered when choosing the framelet basis are its directionality, redundancy and filter size/support. The use of redundant transforms with directional filters is often preferred for image processing applications, because directional transforms (see Chapter 2) capture better transitions in more directions than (bi)orthogonal wavelets [169] (0°, ±45° and 90°). In addition, the redundancy of directional transforms reduces the aliasing introduced by denoising, because this representation provides more "bandwidth" than non-redundant transforms [126]. The last element considered is the filter support/length, since it sets a trade-off between frequency and space localization. For image processing tasks, shorter filters are often preferred because they increase the spatial localization of the transform.

The directionality and compact filters are the main elements that motivate the OHWT as ED path for the noise reduction networks in Chapter 4. However, it can be observed that the redundancy of the OHWT increases the computational cost for its implementation, when compared with non-redundant transforms. Furthermore, the redundancy impacts the model size as well, because redundant transforms generate more bands/feature maps, which increase the number of input/output channels of the denoising network. This is addressed in more detail in Appendix B, where it is shown that the expansive representation provided by the OHWT causes that the number of parameters increases exponentially as the network becomes deeper. Still, it should be noted that this effect can be partly mitigated by sharing parameters within the network, as performed in the DHSN1 and DHSN2 models in Chapter 4, or by using grouped convolutions as implemented in the regularization network shown in Chapter 6. However, it should be noted from the analysis performed in Appendix B, that the models based on the OHWT have more trainable parameters than the models where the encoding-decoding path is learned (the LWFSN). This disadvantage is addressed later.

RQ1b. Is it possible to realize a denoising CNN with a trainable ED path that offers perfect reconstruction, without explicitly training for this property?

The LWFSN presented in Chapter 5 shows that it is possible to learn a model where the ED path approximates perfect reconstruction. There are two main aspects to consider for achieving this property. (1) The LWFSN model is designed in such a way that it can be theoretically proven that it is able to achieve perfect reconstruction, as shown by the analysis performed in Section 5.2.1. (2) The use of shrinkage functions allows a behavior closer to a linear system. Moreover, this type of activation is critical to study the linear behavior of the system, because it becomes linear when the threshold is set to zero. This property enables the impulse and frequency response analyses shown in Section 5.3.

Despite the fact that the LWFSN achieves an ED path where the impulse response is close to the convolution identity, it should be noted that the individual encoding-decoding filter pairs do not comply with such property. A likely explanation for this is that the CNN learns the best filters for *noise reduction*, which may not necessarily guarantee perfect reconstruction for each individual layer, as mentioned in Chapter 3. In addition, effects, such as variations in gain between individual layers, which is known as the *neural balance theorem* [170] may further complicate to study the response of individual layers, because the energy of a specific band may be scaled by other layers in the design.

RQ1c. Can the integration of signal processing knowledge improve CNNs in any aspect?

The use of signal processing concepts such as wavelet shrinkage within CNNs can circumvent some issues associated with conventional ReLU-based models. For example, as shown in Chapter 5, the use of shrinkage activations instead of ReLUs allows to reduce the number of feature maps in CNNs, which inherently reduces the computational cost and complexity the design. Specifically, shrinkage functions propagate the positive and negative elements of the feature maps of the CNN, while ReLUs require twice as many feature maps to propagate the same signal, as mentioned in Section 3.4.1.

In addition, it should be noted that, although not discussed in the technical chapters of this thesis, at the moment of writing this dissertation, other signal processing concepts are being exploited within CNNs and have shown to achieve state-of-the-art image denoising, while remaining interpretable. An example of such integration is the deep total-variation by Kobler *et al.* [114], [171], which integrates elements of total-variation denoising [13], [172] with CNNs.

RQ 2: Impact of the activation functions in CNNs

RQ2a. Do shrinkage functions have any advantages when compared to ReLUs in terms of CNN interpretability?

Shrinkage activations enable more model predictable models when compared with ReLUs, because shrinkage layers only can perform either signal shrinkage, the unity operation or signal boost, whereas the ReLUs can represent any arbitrary function if the number of channels and network depth are large enough [118]. Furthermore, by using shrinkage activations in a CNN, the model can be linearized by setting the threshold to zero, which allows to characterize the behavior of the linear part of the model with the impulse and frequency response of the system. It should be noted that this is not possible with ReLU activations, because they do not become linear when the bias/threshold is set to zero.

RQ2b. How do shrinkage functions limit the operation of CNNs when compared to ReLU activations?

As discussed in RQ2a, ReLUs are universal approximators and can represent functions such as shrinkage and clipping activations (see Section 3.4.2). Consequently, the use of shrinkage layers limits the type of functions that can be represented by the model to shrink/boost or to perform the identity function over the feature maps. This limitation may not be critical for noise reduction applications in which may even have benefits, such as reduced computational complexity. However, shrinkage functions could be a limitation for other more complicated tasks. For example, in image segmentation and image classification, it could be important to preserve only the positive or negative part of a feature map, because it may indicate specific intensity/colour transitions that are relevant for classifying/segmenting a specific pattern. Consequently, we hypothesize that in this context, the use of shrinkage activations may be detrimental for the performance of the model, because shrinkage functions affect equally the positive and negative values of the feature maps.

RQ 3: Comparing characteristics of convolutional neural networks

RQ3a. Is it possible to define a common framework for the analysis of denoising CNNs to identify their potential limitations?

Prior to the work presented in this dissertation, there have been diverse frameworks that have defined a formal analysis of the reconstruction characteristics of CNNs. The analyses performed by these frameworks represent the convolutions and/or down/up-sampling layers of CNNs with matrix-vector notation. This may be confusing for deep learning practitioners that do not have signal processing background, because it assumes that the designer understands how to translate the various elements to an actual implementation, which is not always straightforward. For example, in matrix-vector notation, a multi-resolution wavelet decomposition is represented by a single matrix multiplication. However, in terms of implementation this is equivalent to multiple convolutions and down/up-sampling layers. Consequently, it can be observed that matrix-vector forms hide the local nature of framelet decompositions.

In order to avoid the shortcomings of the vector-matrix form, Section 2.4 and Chapter 3 introduce a notation that reflects better the implementation of the network, which is a clear advantage when compared with matrix-vector representations. In addition, the proposed notation can be easily related to graphical representations as demonstrated by the analyses of Chapters 3, 4 and 5. The key to the improved notation is that the proposed framework models each individual layer in the model, instead of abstractions presented in previous works such as the theory of deep convolutional framelets [40]. This is bridging the gap that deep learning practitioners experience with conventional frameworks.

Finally, a major advantage of the proposed framework is that it explicitly shows the influence of the non-linear part of the model in terms of signal reconstruction. Notably, alternative approaches, such as the theory of deep convolutional framelets [40], only model the linear part of CNNs and ignore the effects of the non-linearities in the reconstruction of the signal.

RQ3b. What are the limitations of such framework?

The main limitations of the analysis framework proposed in Chapter 3 are listed as follows. (1) The analyses presented in Section 3.5 always consider filters that allow for perfect reconstruction, which is a useful property for analysis, but as shown in Section 3.6, trained models do not always comply with such property. (2) The proposed framework does not account for the effects of the number of channels, which impacts the ability of the model to reconstruct the signal. (3) The framework accounts for the effects of the convolution and non-linearities of a given CNN, but does not consider other operations, such as normalization layers. The reason for this omission is that (during test time), these layers perform linear operations, which could be embedded within the convolution weights. Furthermore, their behavior during training is signal-dependent, which greatly complicates any analysis of the model. (4) The final limitation of the framework is that it only studies the signal reconstruction characteristics of CNNs, but it cannot predict the actual noise reduction performance of a given model.

RQ 4: Expanding the applications of encoding-decoding CNNs

RQ4a. Which concepts and principles can be applied to improve the generalization of model-based deep learning?

Chapter 6 presents a model-based CNN for denoising and material decomposition in dual-energy CBCT. The proposed model is referred to as *regularized conjugate-gradient network* (rCGN) and incorporates elements of the CNNs discussed in Chapters 3 and 4. The rCGN model contains two sources of adaptivity that can improve its generalization. (1) The CGLS optimization backbone has adaptive step-size computation updates that are optimized for the material decomposition of every analyzed slice. We hypothesize that this is one of the reasons why the rCGN model generalizes better than FISTA-Net for the regularized material decomposition shown in Chapter 6. More specifically, the trained step-sizes of FISTA-Net assume that all the processed slices generate the same optimization landscape for the material decomposition, whereas the use of the CGLS method ensures that the process is optimized for every processed slice. (2) The thresholds in the regularizer of the rCGN are proportional to the noise intensity of the processed image. It should be noted that we are not the first in using this technique. In fact, this approach has been successfully followed in the multi-scale sparse coding network [48] and it is common in (non-deep learning-based) adaptive filters[14].

RQ4b. Can the proposed concepts with ED CNNs also contribute to lower their complexity and how does this compare with conventional CNN designs?

One of the core elements of the rCGN model is a simplified regularization network that uses a non-trainable framelet as encoding-decoding path. Chapter 6 shows that this simplified CNN produces estimates with slightly lower PSNR and MSSIM values than the reference FISTA-Net, when evaluated in images that are similar to the scans that the network was trained upon. It should be noted that the core of FISTA-Net is a more conventional CNN design, based on ReLU activations. Nevertheless, despite the small degradations in PSNR and MSSIM, the images produced by the proposed rGCN and FISTA-Net are almost identical in terms of appearance. In addition, as shown by the evaluation in clinical scans, the simpler rCGN is able to generalize better to unseen images, such as the clinical scans examined in Chapter 6, thereby improving the robustness of the system.

7.3 Outlook for the integration of signal processing and deep learning

At the moment of writing this dissertation, the use of CNNs that leverage knowledge of signal processing concepts has gained momentum and it is likely that this trend will continue. A possible reason for the increased popularity in model-centric design of CNNs may be related to the fact that the performance of conventional CNNs (often trained and designed with data-centric considerations in mind) is only improving marginally when compared with the initial successes of deep learning approaches. Consequently, properties such as robustness, interpretability and reduced computational complexity have become more relevant and attractive for exploration. This renewed interest in system-centric design of CNNs has sparked the integration of many signal processing concepts that go beyond the framelet interpretation shown in this thesis.

Going beyond the framelet interpretation, parallel developments occur towards alternative deep learning models that integrate established signal processing concepts with CNNs. For example, the *discrete cosine transform 2 network* (DCT2-Net) [119] integrates DCT-based denoising with deep learning. An additional signal processing concept that has gained momentum for its integration with deep learning in image denoising is sparse coding, which has been applied to the *convolutional dictionary learning network* (CDLNet) [173], the *convolutional sparse coding network* (CSC Net) [174], and the *deep K-SVD denoising* [53] model. Finally, the *total deep variation framework* [114], [171] integrates elements of total-variation denoising algorithms [13], [172] with CNNs. It should be noted that all of these models achieve results that are competitive with state-of-the art CNNs, while being more interpretable that conventional models.

In addition to noise reduction, model-based deep learning approaches [139], [175]–[180] have also leveraged signal processing and CNN concepts. (e.g. the DE CBCT material decomposition shown in Chapter 6). This approach is powerful, interpretable and has found success in many problems such as image reconstruction in CT [181] and magnetic resonance imaging (MRI) [160].

It should be noted that the presented examples of CNNs which are used to solve inverse problems such as noise reduction, are only a small sample of the plethora of signal processing-based designs of deep learning models. Due to their better interpretability [58], lower computational complexity [182] and –in some cases– higher performance [114], [173], it is likely that the trend of integrating signal processing concepts with CNNs will continue and may become the *de-facto* approach for solving inverse problems in the coming years. For example, an immediate extension of the work in noise reduction of this dissertation is to employ the proposed models for other similar inverse problems such as image enhancement and creation of super-resolution imagery. This extension is plausible because the models applied to this problem would subjectively enhance the information in the feature maps, rather than suppressing it as in the noise reduction case.

In conclusion, it can be observed that some of the most robust interpretations of CNNs from a signal processing perspective, focus on the context of noise reduction [40], [41], [43]. This is most likely related to the parallels between the mathematical formulation of CNNs and well-known denoising algorithms. However, it should be noted that conventional ED CNNs, such as the U-Net [8], can be used in many pixel-level operations (often without modifications) simply by (re-)training the network with new data. Consequently, the studies presented here can be used as a starting point to understand how ED CNN models operate in other contexts and how this affects the representations that they are learning. However, this is unlikely to happen because the vast majority of deep learning researchers commonly use existing CNN designs without questioning or caring about their internal operation.

APPENDIX A

Errata of "Noise reduction in CT using Learned Wavelet-Frame Shrinkage Networks"

Chapter 5 is based on the article *Noise reduction in CT using Learned Wavelet-Frame Shrinkage Networks* [58]. While editing this thesis, it was found that there is a bug present in the routine that computes the HaarPSI metric [97]. The referred bug causes that the performance measurements in Tables III, IV, V and VI of the referred article are scaled up by a factor of approximately 1.04 times and affects all models. However, it should be noted that the specified mistake does not change the conclusions of the referred publication because it affects all the models in approximately the same way. Furthermore, the HaarPSI is only one out of the three quality metrics displayed in the aforementioned tables. The text below reflects the corrections that have been applied in Chapter 5 of this thesis with respect to published article [58]. Here, Tables A.1, A.2, A.3 and A.4 reflect the applied corrections, in which the incorrect values are presented in red.

Model	HaarPSI [AU]	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
Baseline (QDCT)	0.796	0.835	0.654	25.65
LWFSN	0.836	0.868	0.670	27.19
TF U-Net	0.833	0.874	0.693	27.61
FBPConvNet	0.838	0.876	0.692	27.63

Table A.1. Corrected values of Table III [58]. This dissertation displays the rectifications on Table 5.3.

Model	HaarPSI [AU]	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]
Baseline	0.796	0.835	0.654	25.65
LWFSN	0.836	0.868	0.670	27.19
LWFSN-S	0.833	0.868	0.667	26.99

Table A.2. Corrected values of Table IV [58]. This dissertation displays the rectifications on Table 5.4.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]	
LWFSN	0.835±4.4E-4	0.868±3.2E-4	0.667±1.4E-3	27.12±0.032
LWFSN-S	0.832±1.1E-4	0.867±4.3E-4	0.667±1.0E-3	26.98±0.027

Table A.3. Corrected values of Table V [58]. This dissertation displays the rectifications on Table 5.5.

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]	
LWFSN	0.836	0.868	0.670	27.19
rLWFSN-NRW	0.836	0.868	0.672	27.53
rLWFSN	0.838	0.872	0.670	27.69

Table A.4. Corrected values of Table VI [58]. This dissertation displays the rectifications on Table 5.6.

APPENDIX B

Performance comparison of the proposed models

B.1 Introduction

Chapters 4 and 5 propose two CNNs [57], [58] in which the encoding-decoding paths decompose and reconstruct the signal, while a non-linear section positioned in the middle of the encoding-decoding path eliminates the noise. Specifically, the models are the (residual) wavelet-frame shrinkage network (LWFSN) and the dual-Haar shrinkage networks 1 and 2 (DHSN1 and DHSN2, respectively). A major limitation of the studies presented in Chapters 4 and 5 is that it is not possible to directly compare the performance of the models which they present, since there are differences in their experimental settings. For example, the DHSN1 and DHSN2 models from Chapter 4 use three decomposition levels, while the rLWFSN and LWFSN models from Chapter 5 use four. Furthermore, the training loss, the training loop and learning rate scheduling are different. In order to address this limitation, this appendix compares all the models by training them under similar conditions and by testing their performance based on the same metrics and procedure, as presented in Chapter 5.

B.2 Experiments and results

The experiments presented in this appendix use the same dataset (the low-dose and high-dose CT scans from the cancer imaging archive [183]), data partitioning, optimizer (Adam), training loop, learning rate scheduling and network depth of the experiments in Chapter 5. This means that the DHSN1 and DHSN2 models from Chapter 5 are trained for this experiment with four decomposition levels, instead of the three used in Chapter 4. Furthermore, it was found experimentally that the DHSN1 and DHSN2 models converge on this setting with 300 training epochs, while the training for the (r)LWFSN stays the same as in Chapter 5.

With the above-described settings, the DHSN1, DHSN2, LWFSN and rWFSN models have been trained to estimate full-dose out of quarter-dose CT scans. Based on the data partitioning from Chapter 5, the trained models have been evaluated on the test set and the results are summarized in Table B.1, which presents the performance metrics' outcomes for all models. The measured performance indicators are the Haar

Model	HaarPSI [AU]	MSSIM [AU]	PSNR [dB]	No. params.
Noisy inp.	0.796	0.654	25.65	N/A
rLWFSN	**0.838**	0.670	**27.79**	32,584
LWFSN	0.836	0.670	27.19	32,584
DHSN1	0.837	**0.673**	27.70	420,784
DHSN2	**0.838**	**0.673**	27.70	560,252

Table B.1. Haar PSI, MSSIM and PSNR metrics measured on CT scans from the cancer imaging archive. The table contains the performance measurements of the image estimates produced by the (r)LWFSN and the DHSN1 and DHSN2 models. The method for testing is the same as in Chapter 5 for most of the models and under similar conditions.

perceptual similarity index (Haar PSI), the mean structural similarity index metric (MSSIM) and the peak signal-to-noise ratio (PSNR).

From Table B.1 it can be observed that all the above-described models achieve similar performance. However, it can be observed that the best performing model is the rLWFSN, while the worst is the non-residual LWFSN. The second best model is the DHSN2. In addition to the quantitative performance metrics, Fig. B.1 presents slices processed with each of tested networks, related to Table B.1, where it can be observed that all models perform similarly, although the DHSN1 and DHSN2 tend to produce slightly smoother image estimates than the (r)LWFSN.

In addition to the performance measurements, Table B.1 also displays the total number of trainable parameters for each of the CNNs. It should be noted that the models with the least number of parameters are the rLWFSN and the LWFSN. In contrast, the DHSN2 is the model with the most parameters. This can be understood when considering the exponential growth of feature maps of the encoding-decoding of the DHSN1 and DHSN2, which is the OHWT [57]. As discussed in Section 4.3.3, the number of feature maps of the OHWT increases exponentially with respect to the number of decomposition levels. This issue can be partly circumvented by sharing parameters within the trainable part of the DHSN models.

B.3 Conclusions

The presented experiment shows that the proposed ED CNNs in this thesis are able to produce high-quality full-dose estimates out of low-dose CT to a level which is almost equivalent. The main difference between them is the total number of trainable parameters, which is higher for the DHSN1 and DHSN2 and it is caused by the large number of channels in the high-frequency bands of the OHWT, which is integrated in this model. More specifically, the OHWT produces a large number high-pass channels that are used as input to the denoising CNNs employed in this design. This increases the dimensions of convolution kernels of the CNNs that are used to remove the noise, which results in more parameters than the (r)LWFSN. Finally, it can be observed that the best trade-off between image quality and total number of trainable parameters is provided by the rLWFSN model.

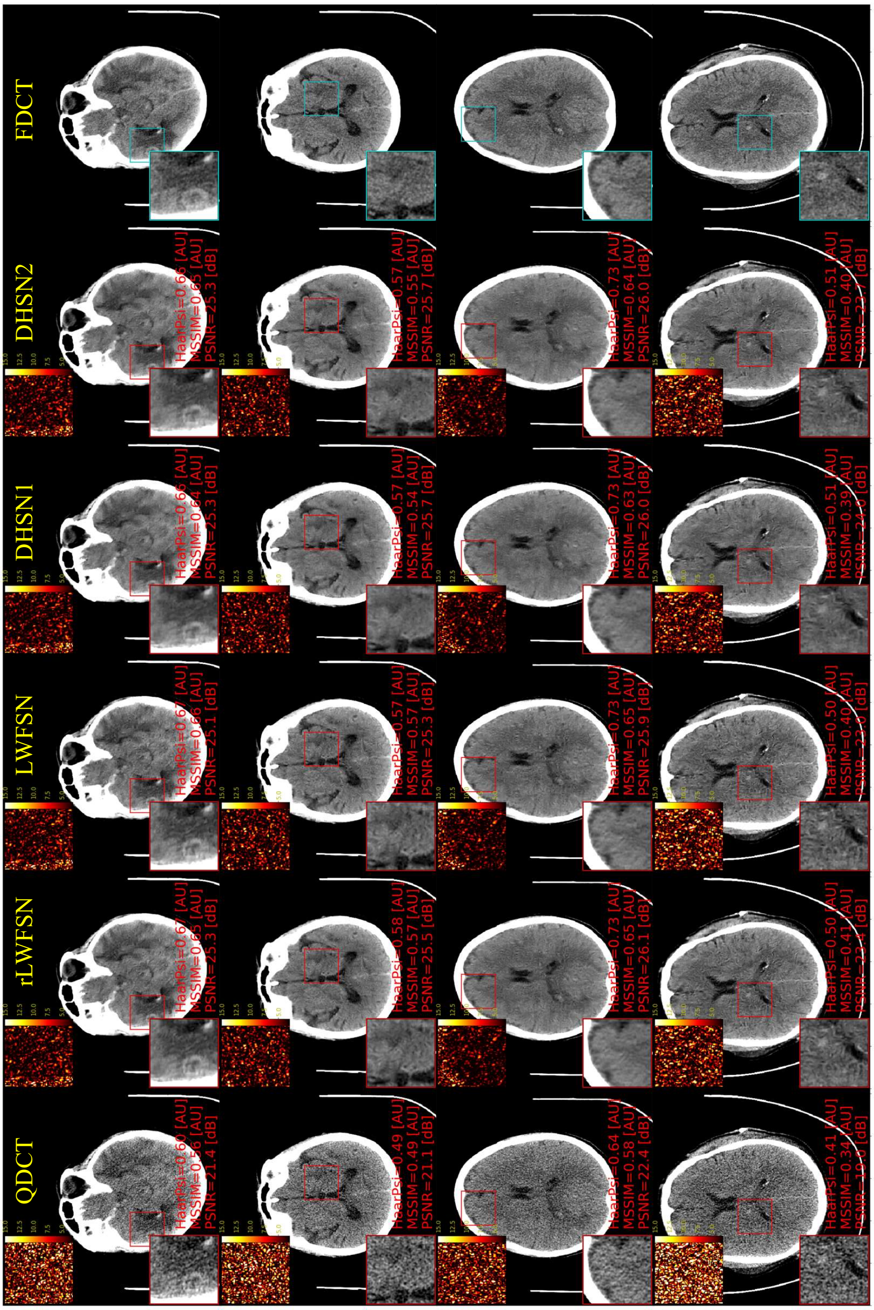


Figure B.1. CT slices processed with the rLWFSN, LWFSN, DNSN1 and DHSN2 models. The figure shows that the smoothest images are produced by the DHSN1 and DHSN2 networks, while the LWFSN is the sharpest-looking model.

Acronyms

CNN Convolutional neural network.

CT Computed tomography.

CBCT Cone-beam computed tomography.

TV Total variation.

ReLU Rectified linear unit.

DoG Derivative of Gaussian.

LET Linear expansion of thresholds.

MAP Maximum a posteriori.

TF U-Net Tight frame U-Net.

RED CNN Residual encoder-decoder CNN.

MSCN Multi-scale sparse coding network.

FBPConvNet Filtered backprojection network.

DTCWT Dual-tree complex wavelet transform.

OHWT Overcomplete Haar wavelet transform.

DHWT Directional hypercomplex wavelet transform.

QWT Quaternion wavelet transform.

DHSN Dual-Haar wavelet transform.

QDCT Quarter-dose CT.

FDCT Full-dose CT.

WSN Wavelet shrinkage network.

NPS Noise power spectrum.

PSNR Peak signal-to-noise ration.

MSSIM Mean structural similarity index metric.

DIP Deep image prior.

LWFSN Learned wavelet frame shrinkage network.

HaarPSI Haar perceptual similarity index.

SGD Steepest gradient descent.

CGLS Conjugate gradient least squares.

DE Dual energy.

CBCT Cone-beam computed tomography.

Bibliography

[1] N. Stiennon, L. Ouyang, J. Wu *et al.*, "Learning to summarize with human feedback", in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 3008–3021. [Online]. Available: `https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf`.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu and M. Chen, "Hierarchical text-conditional image generation with clip latents", *arXiv preprint arXiv:2204.06125*, vol. 1, no. 2, p. 3, 2022.

[3] C. Saharia, W. Chan, S. Saxena *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding", *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.

[4] A. Kirillov, E. Mintun, N. Ravi *et al.*, "Segment anything", *arXiv preprint arXiv:2304.02643*, 2023.

[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[8] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241, ISBN: 978-3-319-24574-4.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising", *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. DOI: `10.1109/TIP.2017.2662206`.

[10] C. Cortes and V. Vapnik, "Support-vector networks", *Machine learning*, vol. 20, pp. 273–297, 1995.

[11] J. Shotton, A. Fitzgibbon, M. Cook *et al.*, "Real-time human pose recognition in parts from single depth images", in *CVPR 2011*, IEEE, 2011, pp. 1297–1304.

[12] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, IEEE, vol. 1, 2001, pp. I–I.

[13] L. I. Rudin, S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[14] S. G. Chang, B. Yu and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression", *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.

[15] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation", in *2006 IEEE Comput. Society Conference on Comput. Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 1, 2006, pp. 895–900.

[16] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[17] M. T. McCann, K. H. Jin and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review", *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017. DOI: 10.1109/MSP.2017.2739299.

[18] S. Y. Khamaiseh, D. Bagagem, A. Al-Alaj, M. Mancino and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification", *IEEE Access*, 2022.

[19] J. Serrà, "When the state of the art is ahead of the state of understanding: Unintuitive properties of deep neural networks", *Metode Science Studies Journal*, no. 9, pp. 127–133, 2019.

[20] N. C. Thompson, K. Greenewald, K. Lee and G. F. Manso, "The computational limits of deep learning", *arXiv preprint arXiv:2007.05558*, 2020.

[21] M. Narwaria, "Explainable machine learning: The importance of a system-centric perspective [lecture notes]", *IEEE Signal Processing Magazine*, vol. 40, no. 2, pp. 165–172, 2023.

[22] R. Couillet, D. Trystram and T. Ménissier, "The submerged part of the ai-ceberg [perspectives]", *IEEE Signal Processing Magazine*, vol. 39, no. 5, pp. 10–17, 2022. DOI: 10.1109/MSP.2022.3182938.

[23] D. Patterson, J. Gonzalez, Q. Le *et al.*, "Carbon emissions and large neural network training", *arXiv preprint arXiv:2104.10350*, 2021.

[24] J. Boice Jr, L. T. Dauer, K. R. Kase, F. A. Mettler Jr and R. J. Vetter, "Evolution of radiation protection for medical workers", *The British Journal of Radiology*, vol. 93, no. 1112, p. 20 200 282, 2020.

[25] L. Yu, X. Liu, S. Leng *et al.*, "Radiation dose reduction in computed tomography: Techniques and future perspective", *Imaging in medicine*, vol. 1, no. 1, p. 65, 2009.

[26] L. A. Killewich and T. A. Singleton, *Governmental regulations and radiation exposure*, 2011.

[27] M. Hilts, A. Jirasek and C. Duzenli, "Technical considerations for implementation of x-ray ct polymer gel dosimetry", *Physics in Medicine & Biology*, vol. 50, no. 8, p. 1727, Apr. 2005. DOI: 10.1088/0031-9155/50/8/008. [Online]. Available: https://dx.doi.org/10.1088/0031-9155/50/8/008.

[28] J. A. Seibert, "Tradeoffs between image quality and dose", *Pediatric radiology*, vol. 34, S183–S195, 2004.

[29] W. Chen, Y. Shao, Y. Wang *et al.*, "A novel total variation model for low-dose CT image denoising", *IEEE Access*, vol. 6, pp. 78 892–78 903, 2018.

[30] A. Borsdorf, R. Raupach, T. Flohr and J. Hornegger, "Wavelet based noise reduction in CT-images using correlation analysis", *IEEE transactions on medical imaging*, vol. 27, no. 12, pp. 1685–1703, 2008.

[31] Y. Chen, L. Shi, Q. Feng *et al.*, “Artifact suppressed dictionary learning for low-dose CT image processing”, *IEEE transactions on medical imaging*, vol. 33, no. 12, pp. 2271–2292, 2014.

[32] S. V. M. Sagheer and S. N. George, “Denoising of low-dose CT images via low-rank tensor modeling and total variation regularization”, *Artificial intelligence in medicine*, vol. 94, pp. 1–17, 2019.

[33] Y. Lei, D. Xu, Z. Zhou *et al.*, “A denoising algorithm for CT image using low-rank sparse coding”, in *Medical Imaging 2018: Image Processing*, SPIE, vol. 10574, 2018, pp. 434–440.

[34] T. Zhao, J. Hoffman, M. McNitt-Gray and D. Ruan, “Ultra-low-dose CT image denoising using modified bm3d scheme tailored to data statistics”, *Medical physics*, vol. 46, no. 1, pp. 190–198, 2019.

[35] L. Chen, S. Gou, Y. Yao, J. Bai, L. Jiao and K. Sheng, “Denoising of low dose CT image with context-based bm3d”, in *2016 IEEE Region 10 Conference (TENCON)*, IEEE, 2016, pp. 682–685.

[36] Y. Han and J. C. Ye, “Framing u-net via deep convolutional framelets: Application to sparse-view ct”, *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1418–1429, 2018.

[37] H. Chen, Y. Zhang, M. K. Kalra *et al.*, “Low-dose ct with a residual encoder-decoder convolutional neural network”, *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2524–2535, 2017.

[38] H. Li, X. Yang, S. Yang, D. Wang and G. Jeon, “Transformer with double enhancement for low-dose CT denoising”, *IEEE Journal of Biomedical and Health Informatics*, pp. 1–12, 2022. DOI: `10.1109/JBHI.2022.3216887`.

[39] F. Fan, M. Li, Y. Teng and G. Wang, “Soft autoencoder and its wavelet adaptation interpretation”, *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1245–1257, 2020. DOI: `10.1109/TCI.2020.3013796`.

[40] J. C. Ye, Y. Han and E. Cha, “Deep convolutional framelets: A general deep learning framework for inverse problems”, *SIAM Journal on Imaging Sciences*, vol. 11, no. 2, pp. 991–1048, 2018.

[41] V. Papyan, Y. Romano and M. Elad, “Convolutional neural networks analyzed via convolutional sparse coding”, *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2887–2938, 2017.

[42] M. Vetterli, “A theory of multirate filter banks”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 356–372, 1987.

[43] J. C. Ye and W. K. Sung, “Understanding geometry of encoder-decoder CNNs”, in *International Conference on Machine Learning*, PMLR, 2019, pp. 7064–7073.

[44] A. Pižurica, “Image denoising algorithms: From wavelet shrinkage to nonlocal collaborative filtering”, *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–17, 2017.

[45] Q. Zhang, Y. Yang, H. Ma and Y. N. Wu, “Interpreting cnns via decision trees”, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6261–6270.

[46] C. Aytekin, “Neural networks are decision trees”, *arXiv preprint*, 2022.

[47] F. Khozeimeh, D. Sharifrazi, N. H. Izadi *et al.*, “Rf-CNN-f: Random forest with convolutional neural network features for coronary artery disease diagnosis based on cardiac magnetic resonance”, *Scientific Reports*, vol. 12, no. 1, pp. 1–12, 2022.

[48] K. Mentl, B. Mailhé, F. C. Ghesu *et al.*, "Noise reduction in low-dose ct using a 3d multiscale sparse denoising autoencoder", in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2017, pp. 1–6.

[49] M. Unser, "A representer theorem for deep neural networks", *Journal of Machine Learning Research*, vol. 20, no. 110, pp. 1–30, 2019. [Online]. Available: `http://jmlr.org/papers/v20/18-418.html`.

[50] K. H. Jin, M. T. McCann, E. Froustey and M. Unser, "Deep convolutional neural network for inverse problems in imaging", *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[51] M. Aharon, M. Elad and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[52] S. S. Chen, D. L. Donoho and M. A. Saunders, "Atomic decomposition by basis pursuit", *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[53] M. Scetbon, M. Elad and P. Milanfar, "Deep k-svd denoising", *IEEE Transactions on Image Processing*, vol. 30, pp. 5944–5955, 2021.

[54] G. Y. Chen, T. D. Bui and A. Krzyżak, "Image denoising with neighbour dependency and customized wavelet and threshold", *Pattern recognition*, vol. 38, no. 1, pp. 115–124, 2005.

[55] L. A. Zavala-Mondragón, P. H. de With and F. van der Sommen, "A signal processing interpretation of noise-reduction convolutional neural networks: Exploring the mathematical formulation of encoding-decoding cnns", *IEEE Signal Processing Magazine*, vol. 40, no. 7, pp. 38–63, 2023. DOI: `10.1109/MSP.2023.3300100`.

[56] L. A. Zavala-Mondragón, P. H.N. de With, D. Ruijters, P. van de Haar and F. van der Sommen, "Dual-energy cbct pre-spectral-decomposition filtering with wavelet shrinkage networks", in *International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2020, pp. 1–6.

[57] L. A. Zavala-Mondragón, P. H. de With and F. van der Sommen, "Image noise reduction based on a fixed wavelet frame and CNNs applied to CT", *IEEE Transactions on Image Processing*, vol. 30, pp. 9386–9401, 2021.

[58] L. A. Zavala-Mondragón, P. Rongen, J. O. Bescos, P. H. de With and F. van der Sommen, "Noise reduction in CT using learned wavelet-frame shrinkage networks", *IEEE Transactions on Medical Imaging*, vol. 41, no. 8, pp. 2048–2066, 2022.

[59] L. A. Zavala-Mondragón, K. J. Engel, P. H. De With and F. Van der Sommen, "Hybrid framelet and CNN approach for denoising and material decomposition in dual-energy CBCT", *In realization*, 2023.

[60] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage", *biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[61] R. R. Coifman and D. L. Donoho, "Translation-invariant de-noising", in *Wavelets and statistics*, Springer, 1995, pp. 125–150.

[62] G. Yu, G. Sapiro and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity", *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481–2499, 2011.

[63] M. Nejati, S. Samavi, H. Derksen and K. Najarian, "Denoising by low-rank and sparse representations", *Journal of Visual Communication and Image Representation*, vol. 36, pp. 28–39, 2016.

[64] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering", *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[65] I. W. Selesnick and K. Y. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms", in *Wavelets: Applications in Signal and Image Processing X*, International Society for Optics and Photonics, vol. 5207, 2003, pp. 607–618.

[66] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging", *Journal of mathematical imaging and vision*, vol. 40, no. 1, pp. 120–145, 2011.

[67] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems", *SIAM journal on imaging sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[68] J.-F. Cai, S. Osher and Z. Shen, "Split bregman methods and frame based image restoration", *Multiscale modeling & simulation*, vol. 8, no. 2, pp. 337–369, 2010.

[69] P. Getreuer, "Rudin-osher-fatemi total variation denoising using split bregman", *Image Processing On Line*, vol. 2, pp. 74–95, 2012.

[70] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising", *SIAM Journal on Scientific Computing*, vol. 17, no. 1, pp. 227–238, 1996. DOI: `10.1137/0917016`. [Online]. Available: `https://doi.org/10.1137/0917016`.

[71] G. Yu and G. Sapiro, "Dct image denoising: A simple and effective image denoising algorithm", *Image Processing On Line*, vol. 1, pp. 292–296, 2011.

[72] F. Luisier, C. Vonesch, T. Blu and M. Unser, "Fast interscale wavelet denoising of poisson-corrupted images", *Signal processing*, vol. 90, no. 2, pp. 415–427, 2010.

[73] G. R. Easley and D. Labate, "Image processing using shearlets", in *Shearlets*, Springer, 2012, pp. 283–325.

[74] N. Kingsbury, "Design of q-shift complex wavelets for image processing using frequency domain energy minimization", in *Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429)*, IEEE, vol. 1, 2003, pp. I–1013.

[75] G. R. Easley, D. Labate and F. Colonna, "Shearlet-based total variation diffusion for denoising", *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 260–268, 2008.

[76] L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency", *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2744–2756, 2002.

[77] K. H. Jin and J. C. Ye, "Sparse and low-rank decomposition of a hankel structured matrix for impulse noise removal", *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1448–1461, 2017.

[78] Y. Zhang, R. Kang, X. Peng *et al.*, "Image denoising via structure-constrained low-rank approximation", *Neural Computing and Applications*, vol. 32, no. 16, pp. 12 575–12 590, 2020.

[79] K. H. Jin, D. Lee and J. C. Ye, "A general framework for compressed sensing and parallel mri using annihilating filter based low-rank hankel matrix", *IEEE Transactions on Computational Imaging*, vol. 2, no. 4, pp. 480–495, 2016.

[80] K. H. Jin and J. C. Ye, "Random impulse noise removal using sparse and low rank decomposition of annihilating filter-based hankel matrix", in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 3877–3881.

[81] C. Szegedy, W. Liu, Y. Jia *et al.*, "Going deeper with convolutions", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[82] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[83] R. Girshick, "Fast r-cnn", in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[84] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[85] V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[86] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Icml*, 2010.

[87] F. Milletari, N. Navab and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation", in *2016 fourth international conference on 3D vision (3DV)*, IEEE, 2016, pp. 565–571.

[88] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks", *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[89] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[90] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.", *Journal of machine learning research*, vol. 11, no. 12, 2010.

[91] A. Makhzani and B. Frey, *K-sparse autoencoders*, 2013. DOI: `10.48550/ARXIV.1312.5663`. [Online]. Available: `https://arxiv.org/abs/1312.5663`.

[92] M. Ranzato, F. J. Huang, Y.-L. Boureau and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition", in *2007 IEEE conference on computer vision and pattern recognition*, IEEE, 2007, pp. 1–8.

[93] G. Lin, C. Shen, A. van den Hengel and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, Jun. 2016.

[94] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks", *arXiv preprint arXiv:1503.02351*, 2015.

[95] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[96] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "Scipy 1.0: Fundamental algorithms for scientific computing in python", *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[97] R. Reisenhofer, S. Bosse, G. Kutyniok and T. Wiegand, "A haar wavelet-based perceptual similarity index for image quality assessment", *Signal Processing: Image Communication*, vol. 61, pp. 33–43, 2018.

[98] L. Zhang, L. Zhang, X. Mou and D. Zhang, "Fsim: A feature similarity index for image quality assessment", *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.

[99] K. H. Jin and J. C. Ye, "Annihilating filter-based low-rank hankel matrix approach for image inpainting", *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3498–3511, 2015.

[100] T. Yokota, H. Hontani, Q. Zhao and A. Cichocki, "Manifold modeling in embedded space: An interpretable alternative to deep image prior", *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[101] K. C. Kusters, L. A. Zavala-Mondragón, J. O. Bescós, P. Rongen, P. H. de With and F. van der Sommen, "Conditional generative adversarial networks for low-dose CT image denoising aiming at preservation of critical image content", in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2021, pp. 2682–2687.

[102] I. Y. Chun, Z. Huang, H. Lim and J. Fessler, "Momentum-net: Fast and convergent iterative neural network for inverse problems", *IEEE Transactions on Pattern Anal. Mach. Intell.*, pp. 1–1, 2020. DOI: `10.1109/TPAMI.2020.3012955`.

[103] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann and M. Unser, "CNN-based projected gradient descent for consistent ct image reconstruction", *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1440–1453, 2018.

[104] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli and C. Fernandez-Granda, "Robust and interpretable blind image denoising via bias-free convolutional neural networks", in *International Conference on Learning Representations*, 2020. [Online]. Available: `https://openreview.net/forum?id=HJlSmC4FPS`.

[105] M. Unser, "From kernel methods to neural networks: A unifying variational formulation", *arXiv preprint arXiv:2206.14625*, 2022.

[106] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling", *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008. DOI: `10.1109/MSP.2007.914731`.

[107] G. Steidl and J. Weickert, "Relations between soft wavelet shrinkage and total variation denoising", in *Joint pattern recognition symposium*, Springer, 2002, pp. 198–205.

[108] T. Blu and F. Luisier, "The sure-let approach to image denoising", *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.

[109] I. Daubechies, M. Defrise and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[110] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding", in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.

[111] J. Xiang, Y. Dong and Y. Yang, "Fista-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging", *IEEE Transactions on Medical Imaging*, vol. 40, no. 5, pp. 1329–1339, 2021. DOI: `10.1109/TMI.2021.3054167`.

[112] J. C. Ye, J. M. Kim, K. H. Jin and K. Lee, "Compressive sampling using annihilating filter-based low-rank interpolation", *IEEE Transactions on Information Theory*, vol. 63, no. 2, pp. 777–801, 2016.

[113] A. Cichocki, R. Zdunek and S.-i. Amari, "Nonnegative matrix and tensor factorization [lecture notes]", *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 142–145, 2008. DOI: `10.1109/MSP.2008.4408452`.

[114] E. Kobler, A. Effland, K. Kunisch and T. Pock, "Total deep variation for linear inverse problems", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7549–7558.

[115] D. Narnhofer, A. Effland, E. Kobler, K. Hammernik, F. Knoll and T. Pock, "Bayesian uncertainty estimation of learned variational mri reconstruction", *IEEE Transactions on Medical Imaging*, 2021.

[116] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", *arXiv preprint arXiv:1207.0580*, 2012.

[117] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[118] I. Daubechies, R. DeVore, S. Foucart, B. Hanin and G. Petrova, "Nonlinear approximation and (deep) relu networks", *Constructive Approximation*, vol. 55, no. 1, pp. 127–172, 2022.

[119] S. Herbreteau and C. Kervrann, "Dct2net: An interpretable shallow CNN for image denoising", *IEEE Transactions on Image Processing*, vol. 31, pp. 4292–4305, 2022.

[120] P. Liu, H. Zhang, K. Zhang, L. Lin and W. Zuo, "Multi-level wavelet-CNN for image restoration", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 773–782.

[121] V. Monga, Y. Li and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing", *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021. DOI: `10.1109/MSP.2020.3016905`.

[122] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool and R. Timofte, "Plug-and-play image restoration with deep denoiser prior", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.

[123] A. Pižurica and W. Philips, "Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising", *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 654–665, 2006. DOI: `10.1109/TIP.2005.863698`.

[124] M. Malfait and D. Roose, "Wavelet-based image denoising using a markov random field a priori model", *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 549–565, 1997.

[125] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[126] I. W. Selesnick, R. G. Baraniuk and N. C. Kingsbury, "The dual-tree complex wavelet transform", *IEEE signal processing magazine*, vol. 22, no. 6, pp. 123–151, 2005.

[127] M. Do and M. Vetterli, "Contourlets: A directional multiresolution image representation", in *Proceedings. International Conference on Image Processing*, vol. 1, 2002, pp. I–I. DOI: `10.1109/ICIP.2002.1038034`.

[128] G. Kutyniok, "Shearlets: From theory to deep learning", *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision*, pp. 1–38, 2021.

[129] P. Fletcher and S. Sangwine, "The development of the quaternion wavelet transform", *Signal Processing*, vol. 136, pp. 2–15, 2017, Hypercomplex Signal Processing, ISSN: 0165-1684. DOI: `https://doi.org/10.1016/j.sigpro.2016.12.025`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0165168416303784`.

[130] J. Adam, C. Nafornita, J.-M. Boucher and A. Isar, "A new implementation of the hyperanalytic wavelet transform", in *2007 International Symposium on Signals, Circuits and Systems*, IEEE, vol. 2, 2007, pp. 1–4.

[131] W. L. Chan, H. Choi and R. G. Baraniuk, "Directional hypercomplex wavelets for multidimensional signal analysis and processing", in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 3, 2004, pp. iii–996.

[132] W. L. Chan, H. Choi and R. G. Baraniuk, "Coherent multiscale image processing using dual-tree quaternion wavelets", *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1069–1082, 2008.

[133] B. Chen, X. Duan, Z. Yu, S. Leng, L. Yu and C. McCollough, "Development and validation of an open data format for ct projection data", *Medical physics*, vol. 42, no. 12, pp. 6964–6972, 2015.

[134] D. Ulyanov, A. Vedaldi and V. Lempitsky, "Deep image prior", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[135] D. O. Baguer, J. Leuschner and M. Schmidt, "Computed tomography reconstruction using deep image prior and learned reconstruction methods", *Inverse Problems*, vol. 36, no. 9, p. 094 004, Sep. 2020. DOI: `10.1088/1361-6420/aba415`. [Online]. Available: `https://doi.org/10.1088%2F1361-6420%2Faba415`.

[136] N. Hurley and S. Rickard, "Comparing measures of sparsity", *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4723–4741, 2009.

[137] J. Baek and N. J. Pelc, "The noise power spectrum in ct with direct fan beam reconstruction", *Medical physics*, vol. 37, no. 5, pp. 2074–2081, 2010.

[138] K. Li, J. Tang and G.-H. Chen, "Statistical model based iterative reconstruction (mbir) in clinical ct systems: Experimental assessment of noise performance", *Medical physics*, vol. 41, no. 4, p. 041 906, 2014.

[139] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Acceleration and convergence", *IEEE Transactions on Image Processing*, vol. 29, pp. 2108–2122, 2019.

[140] R. Rigamonti, A. Sironi, V. Lepetit and P. Fua, "Learning separable filters", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2754–2761.

[141] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[142] F. Luisier, T. Blu and M. Unser, "A new sure approach to image denoising: Interscale orthonormal wavelet thresholding", *IEEE Transactions on image processing*, vol. 16, no. 3, pp. 593–606, 2007.

[143] T. Kwon and J. C. Ye, “Cycle-free cyclegan using invertible generator for unsupervised low-dose CT denoising”, *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1354–1368, 2021. DOI: 10.1109/TCI.2021.3129369.

[144] A. Paszke, S. Gross, F. Massa *et al.*, “Pytorch: An imperative style, high-performance deep learning library”, in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[145] T. R. Moen, B. Chen, D. R. Holmes III *et al.*, “Low dose ct image and projection dataset”, *Medical Physics*, 2020.

[146] K. Clark, B. Vendt, K. Smith *et al.*, “The cancer imaging archive (tcia): Maintaining and operating a public information repository”, *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[147] H. Radhiana, S. Syazarina, M. S. Azura, H. Hilwati and M. Sobri, “Non-contrast computed tomography in acute ischaemic stroke: A pictorial review”, *Med J Malaysia*, vol. 68, no. 1, pp. 93–100, 2013.

[148] P. B. Gorelick, F. D. Testai, G. J. Hankey and J. M. Wardlaw, *Hankey’s clinical neurology*. CRC Press, 2020.

[149] A. W. Toga, A. W. Toga, J. C. Mazziotta and J. C. Mazziotta, *Brain mapping: the methods*. Academic press, 2002, vol. 1.

[150] Y. Chun and J. A. Fessler, “Deep bcd-net using identical encoding-decoding CNN structures for iterative image recovery”, in *2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, 2018, pp. 1–5. DOI: 10.1109/IVMSPW.2018.8448694.

[151] C. H. McCollough, S. Leng, L. Yu and J. G. Fletcher, “Dual-and multi-energy ct: Principles, technical approaches, and clinical applications”, *Radiology*, vol. 276, no. 3, pp. 637–653, 2015.

[152] K. M. Brown, S. Zabic and G. Shechter, “Impact of spectral separation in dual-energy ct with anti-correlated statistical reconstruction”, *Proc. 13th fully 3D image recons. in radiol. and nuclear med.*, pp. 491–494, 2015.

[153] T. Niu, X. Dong, M. Petrongolo and L. Zhu, “Iterative image-domain decomposition for dual-energy ct”, *Medical physics*, vol. 41, no. 4, p. 041 901, 2014.

[154] W. Zhang, H. Zhang, L. Wang *et al.*, “Image domain dual material decomposition for dual-energy ct using butterfly network”, *Medical physics*, vol. 46, no. 5, pp. 2037–2051, 2019.

[155] N. Shlezinger, J. Whang, Y. C. Eldar and A. G. Dimakis, “Model-based deep learning”, *Proceedings of the IEEE*, vol. 111, no. 5, pp. 465–499, 2023. DOI: 10.1109/JPROC.2023.3247480.

[156] Z. Li, Y. Long and I. Y. Chun, “An improved iterative neural network for high-quality image-domain material decomposition in dual-energy ct”, *Medical Physics*, 2022.

[157] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[158] N. Parikh and S. Boyd, *Proximal algorithms. number 3 in foundations and trends in optimization*, 2013.

[159] F. Wen, Z. Zhang and Y. Liu, "Generalized thresholding for low-rank tensor recovery: Approaches based on model and learning", in *Tensors for Data Processing*, Elsevier, 2022, pp. 121–151.

[160] H. K. Aggarwal, M. P. Mani and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems", *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2018.

[161] W. A. Kalender, W. H. Perman, . . Vetter and E. Klotz, "Evaluation of a prototype dual-energy computed tomographic apparatus. i. phantom studies", *Medical physics*, vol. 13, no. 3, pp. 334–339, 1986.

[162] S. Kawata and O. Nalcioglu, "Constrained iterative reconstruction by the conjugate gradient method", *IEEE transactions on medical imaging*, vol. 4, no. 2, pp. 65–71, 1985.

[163] M. Simon, K. J. Engel, B. Menser, W. Ruetten, A. Thran and D. Schaefer, "Physical image simulation of human brain in case of acute stroke", in *Medical Imaging 2020: Physics of Medical Imaging*, International Society for Optics and Photonics, vol. 11312, 2020, p. 1 131 234.

[164] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike and A. C. Evans, "Brainweb: Online interface to a 3d mri simulated brain database", *NeuroImage*, vol. 5, 425–Proceedings of 3rd International Conference on Functional Mapping of the Human Brain, 1997.

[165] K. J. Engel, B. Menser, P. Rohr, W. Ruetten, M. Simon and A. Thran, "Dual layer x-ray detector simulation", *Proc. SPIE*, vol. 11312, 113121O, 2020.

[166] W. Van Aarle, W. J. Palenstijn, J. Cant *et al.*, "Fast and flexible x-ray tomography using the astra toolbox", *Optics express*, vol. 24, no. 22, pp. 25 129–25 147, 2016.

[167] X. Wang, K. Yu, S. Wu *et al.*, "Esrgan: Enhanced super-resolution generative adversarial networks", in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.

[168] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, "Residual dense network for image super-resolution", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.

[169] D. Recoskie and R. Mann, "Learning filters for the 2d wavelet transform", in *2018 15th Conference on Computer and Robot Vision (CRV)*, IEEE, 2018, pp. 198–205.

[170] R. Parhi and R. D. Nowak, "Deep learning meets sparse regularization: A signal processing perspective", *IEEE Signal Processing Magazine*, vol. 40, no. 6, pp. 63–74, 2023. DOI: 10.1109/MSP.2023.3286988.

[171] E. Kobler, A. Effland, K. Kunisch and T. Pock, "Total deep variation: A stable regularization method for inverse problems", *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 12, pp. 9163–9180, 2021.

[172] A. Chambolle, "An algorithm for total variation minimization and applications", *Journal of Mathematical imaging and vision*, vol. 20, no. 1, pp. 89–97, 2004.

[173] N. Janjušević, A. Khalilian-Gourtani and Y. Wang, "Cdlnet: Noise-adaptive convolutional dictionary learning network for blind denoising and demosaicing", *IEEE Open Journal of Signal Processing*, vol. 3, pp. 196–211, 2022.

[174] D. Simon and M. Elad, "Rethinking the csc model for natural images", *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[175] J. Sun, H. Li, Z. Xu *et al.*, "Deep admm-net for compressive sensing mri", *Advances in neural information processing systems*, vol. 29, 2016.

[176] U. S. Kamilov, H. Mansour and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems", *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1872–1876, 2017.

[177] S. Ravishankar, I. Y. Chun and J. A. Fessler, "Physics-driven deep training of dictionary-based algorithms for mr image reconstruction", in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 1859–1863. DOI: 10.1109/ACSSC.2017.8335685.

[178] N. Shlezinger and T. Routtenberg, "Discriminative and generative learning for the linear estimation of random signals [lecture notes]", *IEEE Signal Processing Magazine*, vol. 40, no. 6, pp. 75–82, 2023.

[179] W. Xia, H. Shan, G. Wang and Y. Zhang, "Physics-/model-based and data-driven methods for low-dose computed tomography: A survey", *IEEE Signal Processing Magazine*, vol. 40, no. 2, pp. 89–100, 2023.

[180] U. S. Kamilov, C. A. Bouman, G. T. Buzzard and B. Wohlberg, "Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications", *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 85–97, 2023.

[181] W. Xia, Z. Lu, Y. Huang *et al.*, "Magic: Manifold and graph integrative convolutional network for low-dose CT reconstruction", *IEEE Transactions on Medical Imaging*, 2021.

[182] Y. Guo, A. Davy, G. Facciolo, J.-M. Morel and Q. Jin, "Fast, nonlocal and neural: A lightweight high quality solution to image denoising", *IEEE Signal Processing Letters*, vol. 28, pp. 1515–1519, 2021.

[183] C. McCollough, B. Chen, D. Holmes *et al.*, "Low dose ct image and projection data [data set]", *The Cancer Imaging Archive*, vol. 10, 2020.

Publication list

Articles in peer-reviewed technical magazines

[M1] **L. A. Zavala-Mondragón**, P. H. De With and F. Van der Sommen, “A signal processing interpretation of noise-reduction convolutional neural networks: Exploring the mathematical formulation of encoding-decoding CNNs”, *IEEE Signal Processing Magazine*, vol. 40, no. 7, pp. 38–63, 2023.

Articles in peer-reviewed journals

[J1] **L. A. Zavala-Mondragón**, K. J. Engel, B. Menser, F. Ståhl, P. H. De With and F. Van der Sommen, “Hybrid framelet and CNN approach for denoising and material decomposition in dual-energy CBCT”, *IEEE Transactions on Medical Imaging*, (Submitted), 2023.

[J2] **L. A. Zavala-Mondragón**, P. Rongen, J. O. Bescos, P. H. de With and F. van der Sommen, “Noise reduction in CT using learned wavelet-frame shrinkage networks”, *IEEE Transactions on Medical Imaging*, vol. 41, no. 8, pp. 2048–2066, 2022.

[J3] **L. A. Zavala-Mondragón**, P. H. de With and F. van der Sommen, “Image noise reduction based on a fixed wavelet frame and CNNs applied to CT”, *IEEE Transactions on Image Processing*, vol. 30, pp. 9386–9401, 2021.

Articles in international peer-reviewed conferences

[C1] **L. A Zavala-Mondragón.**, F. van der Sommen *et al.*, “On the performance of learned and fixed-framelet shrinkage networks for low-dose CT denoising”, in *Medical Imaging with Deep Learning*, MIDL, 2022.

[C2] **L. A. Zavala-Mondragón**, K. J. Engel, B. Menser, D. Ruijters, F. van der Sommen *et al.*, “Iterative reconstruction anti-correlated ROF model for noise reduction in dual-energy CBCT imaging”, in *Medical Imaging 2021: Image Processing*, SPIE, vol. 11596, 2021, pp. 661–670.

[C3] K. C. Kusters, **L. A. Zavala-Mondragón**, J. O. Bescós, P. Rongen, P. H. de With and F. van der Sommen, “Conditional generative adversarial networks for low-dose CT image denoising aiming at preservation of critical image content”, in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2021, pp. 2682–2687.

[C4] **L. A. Zavala-Mondragón**, D. Ruijters, P. van de Haar, P. H. de With and F. van der Sommen, “Dual-energy CBCT pre-spectral-decomposition filtering with wavelet shrinkage networks”, in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2020, pp. 1–6.

[C5] **L. A. Zavala-Mondragón**, F. van der Sommen, D. Ruijters, K. J. Engel, H. Steinhauser and P. H. de With, “Robust algorithm for denoising of photon-limited dual-energy cone beam CT projections”, in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2020, pp. 867–871.

Acknowledgements

I would like to begin this acknowledgement by evoking a quote by Newton: "If I have seen further it is by standing on the shoulders of Giants". This phrase is often used for describing the incremental nature of scientific development. However, I think that there have been many giants on my way that have allowed me to see diverse aspects of life and science that have guided my way until (and perhaps after of) this research.

To start, I thank my promotor Prof. Peter H.N. de With and my co-promotor Dr. Fons van der Sommen who trusted me to develop research within the VCA group and who guided me during the development of this doctoral project.

With Peter, I spend many (really many) hours reviewing this thesis as well as the scientific articles that compose it. I have to say that in spite of his very busy schedule, Peter always was kind to me, took me seriously and went the extra mile to help me to write papers that were mathematically consistent and with a clear story. This hard work did not go unnoticed, we received comments on the good writing in the rebuttals of the articles that we submitted during this PhD. I think that this was one of the key elements in achieving the acceptance of the most important articles that we published during this project.

To Fons I thank him for trusting me with the NEXIS project and for his guidance and feedback during its realization. In addition, I would like to thank him for always encourage me to aim higher. Without his encouragement, I would have not even considered to write articles for the high-impact factor journals where we successfully published. Furthermore, I want to express my gratitude to him as well as to Dr. Ruud van Sloun for helping me to start an academic career by hiring me for the postdoc position within the AI FORSchung project. I hope this is the beginning of a fruitful collaboration.

I also would like to thank deeply my partner Marjolein who supported me (and tolerated me) during the realization of this project. In addition, she always helped me to see the good things of the PhD when I was very stressed. Furthermore, I also would like to thank my daughter, who helped me to accelerate the review process by opening time into everyone's agenda.

I am very grateful with my parents Rosalba and Juan for their loving support, which allowed me to focus full time in my studies. Furthermore, being the youngest of a large family of talented people has been a great privilege that has allowed me to have a broader view of things. For example, from Lizbeth I learned a lot about writing and that, besides scientific skills there are other attributes that make us humans and that we should also cultivate. Fabiola teached me how to formulate and solve mathematical problems and to enjoy a well-written technical text book. Karen designed the cover of this thesis and has always supported me. I thank Claudia for her support and for her

down-to-earth personality. Finally, I express my gratitude to my brother Socrates for his warmth and for always supporting me and taking care of me.

I also want to thank my friends for their support, specially to Iris, Alessio, Ilaria, Andra, Chris, Julián, Sauvik, Luca, Moises, Vanessa and Lucas, Deodato, Michelle, Areli, Eduardo, Oscar, Jesús, Mario, Nohemí, Luis (Tipo), Gladys, Sylvia, Chucho, Aldo and Nubia. With all of them I've shared many great moments along my stay in the Netherlands and in Mexico.

I thank as well to my colleagues from the VCA group with whom I shared the PhD struggles, had great technical discussions and a lot of fun.

I also express my gratitude to Anja, who planned most of the meetings necessary to review this thesis, and even accelerated them after we knew that my daughter was coming. In addition, she was always very kind to me and planned many social events for us. I hope she has a good recovery.

I would like to express my gratitude of my collaborators of the NEXIS project. Furthermore, I thank as well as the European Union for the funding for performing this research through the Horizon 2020 Grant No. 780026 "NEXIS".

My sincere thanks to my collaborators at AI FORSchung Ahmet, Ruud and Fons for their help and understanding during this time.

I thank as well to Prof. Gerard de Haan, Bishal, Andreia and Wenjin for guiding my first steps in computer vision and teach me to write research papers.

To conclude, I express my gratitude to my high-school teacher, the physicist Mauricio Bravo, without whom I would not have pursued an engineering career. From him I learned the scientific method, to make a mathematical model and the admiration for great scientists. I hope he is still inspiring many students as he did with me.

Curriculum vitae

Luis Albert Zavala Mondragón was born in Naucalpan, México in 1990. In 2014, he obtained a bachelor degree in electrical engineering by the Universidad Nacional Autónoma de México (UNAM), where he developed interest in medical imaging and hardware design. Afterwards, Luis worked as hardware emulation engineer at Intel in Guadalajara, Mexico (2014-2015).

In 2015, Luis was awarded the ALSP scholarship to study the master in electrical engineering at the Eindhoven University of Technology (TU/e) in the Netherlands. There he developed understanding on signal processing, statistics and deep learning. During his study, Luis was intern at Philips research (2016-2017), where he participated in two research projects of computer vision and signal processing for medical applications. Both of these projects resulted into publications in peer-reviewed journals. After graduating from TU/e, Luis joined the medical startup Thirona (Nijmegen, the Netherlads) where he further developed skills in deep learning and CT imaging (2018-2019).

In 2019, Luis joined the Video Coding & Architectures group (VCA) at TU/e as PhD candidate within the European NEXIS project, where he developed noise reduction algorithms for CT images. Two of these algorithms are published in the journals *IEEE Transactions on Medical Imaging* and *IEEE Transactions on Image Processing*. In addition, his insights on the internal operation of noise reduction CNNs are summarized in the article "A signal processing interpretation of noise-reduction CNNs", which is published in the *IEEE Signal Processing Magazine*. At present, Luis continues working in the VCA lab in the European AI FORSchung project, which explores signal processing and generative modeling to exploit the massive data streams generated by fiber optic sensing.